



UNIVERSIDAD NACIONAL DE INGENIERIA
FACULTAD DE ELECTROTECNIA Y COMPUTACION

Diseño e Implementación de un Taxímetro Digital
Multipasajeros

Elaborado por:

Br. Edgard Danilo Avendaño Barrios.

Br. Kelvin Josué Olivares Ruiz.

Tutor: Ing. Camilo Roosevelt Lindo Carrión.

Managua, Nicaragua; 07 de Junio de 2016

DEDICATORIA

Edgard Danilo Avendaño Barrios

La presente tesis va dedicada a nuestro Dios por permitirnos llegar a este punto en nuestras vidas. A mis padres, Adela Barrios y Efraín Avendaño, por darme el apoyo total durante mis años de estudiante universitario, por estar a mi lado siempre alentándome a continuar en este camino. A mi familia que me ha brindado su mano amiga para estar siempre constante en mis estudios. A nuestro tutor, Camilo Lindo Carrión, por facilitarnos aquellos recursos y conocimientos necesarios para la finalización correcta de esta monografía; por sus consejos y directrices hacia las cuales nos movimos para tener éxito en esta labor. Y a todas las personas que estuvieron presentes durante mi carrera profesional, que hicieron posible a mi persona ser mejor cada día.

Kelvin Josué Olivares Ruiz

Este trabajo está dedicado principalmente a nuestro Señor Jesucristo que me ha permitido la vida, salud y capacidad para culminar con éxito mi carrera universitaria y pasar a una nueva etapa de la vida. También va dedicado para mis padres, Norma Ruiz y Oscar Olivares, y a mi familia en general, que me han brindado el apoyo emocional e instructivo a lo largo de los años requeridos para la culminación de mis estudios universitarios. Otra persona a la cual dedico este esfuerzo es a mi novia y pareja durante todos mis años universitarios, María Santos Blanco, pues es la que me ha empujado a seguir adelante cada día e ir más allá de lo que creo que son mis limitaciones. A nuestro tutor, Camilo Lindo Carrión, por llevarnos por el camino correcto para la culminación de nuestros estudios. Y por último, dedico este trabajo a todas las personas que han hecho posible que llegue hasta este punto; a todos y a cada uno: Muchas gracias.

RESUMEN

El presente documento presenta los pasos que se siguieron para realizar el diseño y elaboración del proyecto: Taxímetro Digital Multipasajeros. El mismo está enfocado al departamento de Managua y se ha desarrollado basado en los requerimientos del ente regulador IRTRAMMA y en la Ley 524: Ley General de Transporte Terrestre.

Este taxímetro tiene la capacidad de fijar los precios de hasta un máximo de cuatro carreras distintas simultáneas. Fue diseñado de esta manera porque los taxis en Managua no son exclusivos, sino que realizan varias carreras a la vez siempre que sean sobre la misma ruta considerando un máximo de cuatro carreras teniendo en cuenta la capacidad máxima del vehículo. De igual forma plantea un modelo tarifario hasta ahora inexistente en nuestro país.

El modelo tarifario propuesto asigna adecuadamente los costos, establece un criterio de comparación de costos, parámetros e indicadores y determina técnicamente los valores estándar máximos y mínimos de los parámetros operativos del servicio basado en datos reales proporcionados por la entidad reguladora IRTRAMMA.

En la primera parte de este documento se encuentran los motivos por los cuales este proyecto fue realizado y también los objetivos que se plantearon inicialmente para atacar el asunto.

En la segunda parte se encuentran los fundamentos teóricos en los que se sustenta el diseño del prototipo, empezando desde una breve historia del taxímetro, sus partes y su funcionamiento, la tecnología GPS como auxiliar para la medición de la distancia, el microcontrolador como cerebro del taxímetro, una explicación detallada del microcontrolador usado, el ATmega328P, la comunicación USB usando como puente el microcontrolador ATmega8U2, y

culmina con las diferentes maneras con las que se ha tratado de regular el transporte urbano selectivo en nuestro país.

En la tercera parte se presenta el diseño del prototipo (hardware, software, y empaquetado) de manera detallada para una rápida comprensión de cómo se llevó a cabo la elaboración del proyecto.

En la cuarta parte se presentan los resultados obtenidos en las pruebas realizadas en campo con el taxímetro construido, mostrando el desempeño reflejado por el dispositivo en un escenario real.

Y en la quinta y última parte se presentan las conclusiones luego de haber realizado las pruebas en campo del proyecto y las recomendaciones como propuestas para la mejora del diseño presentado.

INDICE DE CONTENIDOS

1	INTRODUCCIÓN	1
2	OBJETIVOS	3
2.1	Objetivo General.	3
2.2	Objetivos Específicos.	3
3	JUSTIFICACIÓN	4
4	MARCO TEÓRICO	5
4.1	Regulación del Sector Taxi en Managua	5
4.2	Qué es un Taxímetro	7
4.2.1	Consideraciones en un Taxímetro	8
4.2.2	Funcionamiento de un Taxímetro	9
4.2.3	El Taxímetro y sus Partes	10
4.3	GPS (Global Positioning System)	11
4.3.1	Segmento Espacial:	11
4.3.2	Segmento de Control:	13
4.3.3	Segmento Usuario:	14
4.3.4	Funcionamiento del GPS	15
4.3.5	Protocolo NMEA 0183	17
4.3.6	Sistema de Coordenadas Geográficas	18
4.4	Comunicación USB	19
4.4.1	Clases de USB	20
4.5	El Microcontrolador (μ C)	21
4.5.1	Microcontrolador ATmega 328P	23
4.5.1.1	Memorias del Microcontrolador	24
4.5.1.2	Líneas I/O Programables	25
4.5.1.3	Interrupciones Externas e Internas:	26
4.5.2	Microcontrolador ATmega8U2	26
4.5.2.1	Controlador USB:	28
5	DISEÑO DEL SISTEMA	29
5.1	Modelo Tarifario	29
5.1.1	Costos Fijos	29
5.1.2	Costos Variables	30
5.1.3	Cálculo De La Tarifa	32
5.1.3.1	Número De Días En Operación Por Año.	33
5.1.3.2	Jornada Diaria del Taxista.	34
5.1.3.3	Número Promedio de Viajes por Día.	34
5.1.3.4	Kilómetros Diarios con Pasajeros	35
5.1.3.5	Kilómetros Totales Diarios	35
5.1.4	Ecuación del Cálculo de la Tarifa	35
5.1.4.1	Tarifa Inicial	35

5.1.4.2	Tarifa Variable por Distancia	36
5.1.4.3	Tarifa Total.	37
5.1.4.4	Tarifa Extraordinaria	38
5.2	Diseño del Prototipo	38
5.2.1	Diseño de Hardware	39
5.2.1.1	Altium Designer Versión 13	40
5.2.1.2	ATMega328P	42
5.2.1.3	Módulo GPS	43
5.2.1.4	Pantalla LCD	45
5.2.1.5	Selección de la Impresora	50
5.2.1.6	Comunicación USB	51
5.2.1.7	Diseño de la Fuente de Alimentación	53
5.2.1.8	Diseño de la Placa de Circuito Impreso PCB	55
5.2.2	Desarrollo de Firmware	58
5.2.2.1	Atmel Studio 6.2	59
5.2.2.2	Estado Configuración del Sistema	60
5.2.2.3	Estado Libre	62
5.2.2.4	Menú Agregar Pasajero	63
5.2.2.5	Función Principal (Pasajero a Bordo)	66
5.2.2.6	Menú Terminar Carrera	71
5.2.2.7	Medición de Distancia	74
5.2.2.8	Cálculo de la Tarifa.	75
5.2.2.9	Programación De Los Microcontroladores	75
5.2.3	Interfaz en LabView	77
5.2.4	Diseño de Empaquetado	78
6	ANÁLISIS DE RESULTADOS	80
6.1	Prueba Funcional	80
6.1.1	Recepción y Decodificación de Datos del GPS	81
6.1.2	Verificación de Datos en Estado Libre	82
6.1.3	Funcionalidad en Menú Agregar Pasajero	82
6.1.4	Funcionalidad en Función Principal	84
6.2	Pruebas de Impresión	86
6.3	Pruebas De Distancia	87
6.4	Pruebas de Costos por Carrera	88
6.5	Análisis Económico	92
7	CONCLUSIONES	95
8	RECOMENDACIONES	97
9	BIBLIOGRAFIA	99
10	REFERENCIAS	101

INDICE DE FIGURAS

<i>Ilustración 1- Taxímetro Común</i>	<i>8</i>
<i>Ilustración 2- Diagrama en Bloques del Taxímetro Común</i>	<i>10</i>
<i>Ilustración 3- Funcionamiento GPS.</i>	<i>11</i>
<i>Ilustración 4- Representación del Proceso de Trilateración</i>	<i>15</i>
<i>Ilustración 5- Trama NMEA 0183</i>	<i>18</i>
<i>Ilustración 6- Componentes Comunes del Microcontrolador</i>	<i>22</i>
<i>Ilustración 7- Atmega328P-PU.</i>	<i>23</i>
<i>Ilustración 8- Memoria Flash</i>	<i>24</i>
<i>Ilustración 9- Memoria SRAM</i>	<i>25</i>
<i>Ilustración 10- ATmega8U2-Au</i>	<i>27</i>
<i>Ilustración 11- Conexión USB Atmega8u2.</i>	<i>28</i>
<i>Ilustración 12- Desarrollo de Hardware</i>	<i>40</i>
<i>Ilustración 13- Editor de Esquemáticos Altium Designer</i>	<i>41</i>
<i>Ilustración 14- Distribución de Pines ATmega328P</i>	<i>43</i>
<i>Ilustración 15- Módulo GPS</i>	<i>44</i>
<i>Ilustración 16- Módulo LCD 20x4</i>	<i>46</i>
<i>Ilustración 17- Conexiones LCD</i>	<i>49</i>
<i>Ilustración 18- Mini impresor CSN-A2</i>	<i>50</i>
<i>Ilustración 19- Conversión USB TTL</i>	<i>52</i>
<i>Ilustración 20- Fuente de Alimentación</i>	<i>54</i>
<i>Ilustración 21- PCB del Taxímetro en el Editor PCB de Altium Designer.</i>	<i>55</i>
<i>Ilustración 22- Vistas Frontal en 3D de la PCB.</i>	<i>57</i>
<i>Ilustración 23- Vistas Frontal y Trasera en 3D de la tarjeta USB.</i>	<i>57</i>
<i>Ilustración 24- Diagrama de Flujos del Firmware</i>	<i>58</i>
<i>Ilustración 25- Atmel Studio 6.</i>	<i>60</i>
<i>Ilustración 26- Diagrama de Flujos Estado Configuración del Sistema</i>	<i>61</i>
<i>Ilustración 27- Diagrama de Flujos Estado Libre</i>	<i>62</i>
<i>Ilustración 28- Diagrama de Flujos Menú Agregar Pasajero</i>	<i>63</i>
<i>Ilustración 29- Diagrama de Flujos Función Principal</i>	<i>67</i>
<i>Ilustración 30- Diagrama de Flujos Menú Terminar Carrera</i>	<i>72</i>

Ilustración 31- Pines Conector ICSP	76
Ilustración 32- Programador USBTinyISP.	76
Ilustración 33- Interfaz Visual en LabView.	77
Ilustración 34- Instrumento Virtual de la Interfaz (VI).	78
Ilustración 35- Diseño Empaquetado en SketchUp	79
Ilustración 36- Adaptador Universal de Alimentación para Autos.	80
Ilustración 37- Pantalla Cargando Configuración	81
Ilustración 38- Pantalla Estado Libre	82
Ilustración 39- Pantalla Menú Agregar Pasajero	83
Ilustración 40- Pantalla Menú Agregar Pasajero/Imprimir Ticket	83
Ilustración 41- Pantalla Menú Agregar Pasajero/Pasajero Agregado	83
Ilustración 42- Pantalla Función Principal	84
Ilustración 43- Pantalla Menú Terminar Carrera	85
Ilustración 44- Pantalla Menú Terminar Carrera/Imprimir Recibo	85
Ilustración 45- Pantalla Menú Terminar Carrera/A Pagar	86
Ilustración 46- Prueba de Impresión/Ticket Identificativo	86
Ilustración 47- Prueba de Impresión/Recibo	87

INDICE DE TABLAS

Tabla 1- Costos Fijos en el Servicio Taxi	30
Tabla 2- Costos Variables en el Servicio Taxi	32
Tabla 3- Días Feriados Oficiales	33
Tabla 4- Días Laborales en el Año	34
Tabla 5- Detalle de Aprovechamiento Diario del Servicio Taxi	35
Tabla 6- Distribución de Pines del Módulo LCD	48
Tabla 7- Consumo Detallado de Corriente del Dispositivo	54
Tabla 8- Asignación de Fichas Identificativas	64
Tabla 9- Pruebas de Tiempo de Conexión GPS en Carretera Abierta	81
Tabla 10- Pruebas de Tiempo de Conexión GPS en Carretera Abierta	88
Tabla 11- Resultados Recorrido Linda Vista-UCA	89
Tabla 12- Resultados Recorrido UCA-7Sur	90
Tabla 13- Resultados Plaza Inter-Salvador Allende	91
Tabla 14- Costo Total PCB Principal	93
Tabla 15- Costo Total PCB de Programación USB	94

1 INTRODUCCIÓN

El servicio de taxis, es un servicio público preponderante para el desarrollo económico del país, el cual está sujeto a regulación a través de entidades estatales. La regulación económica, es decir la fijación de tarifa por prestación de servicio en la capital, es competencia del Instituto Regulador de Transporte del Municipio de Managua IRTRAMMA, regulación que, a priori no se ha realizado.

La falta de regulación económica en el sector taxi que ha sido motivo de inconformidad en el pueblo nicaragüense debido a las altas tarifas que son aplicadas por los distintos choferes y que no se corresponden con lo que debería ser el precio justo según el criterio del pueblo. Gran parte de la población está en común acuerdo en que las considerables bajas que ha sufrido el precio del combustible, deberían ser un indicativo de gran peso en la fijación de precios y tarifas, hecho que en la realidad no sucede así. ¹

El valor que le es dado a cada carrera realizada por un taxi es basado en lo que el chofer considere apropiado según su propia experiencia y el cliente puede “regatear” este valor. Consciente de la importancia de este rubro en la economía del país, es intolerante que la fijación de los precios sea “experimental” y no técnica, hecho que perjudica tanto al usuario como al taxista.

El taxímetro es el dispositivo electrónico que fija la tarifa de cada carrera realizada por un taxi, siendo éste el regulador económico por excelencia². Sin embargo el taxímetro ha sido debatido y descartado hasta el momento debido a que en nuestro país se utiliza el mismo vehículo para varios usuarios de manera

¹ www.laprensa.com.ni/2014/12/07/economia/1437051-gasolinas-caen-mas-del-25-gasolinas-caen-mas-del-25

² m.diccionario.motorgiga.com/diccionario/taximetro-definicion-significado/gmx-niv15-con195692.htm

simultánea, hecho que el taxímetro común no puede efectuar ya que éste es exclusivo del pasajero que lo solicitó.

Considerando la situación de inconformidad de la población nicaragüense en lo referente a las tarifas en el sector taxi, este documento aborda el diseño de un taxímetro diferente al común, que servirá como respuesta a esta necesidad. La mejora que salta a la vista en este taxímetro es la capacidad de poder realizar más de una carrera a la vez (cuatro posibles carreras simultáneas e independientes entre sí). Con la regulación adecuada y con el diseño de mucha precisión mostrado en el documento se pretende lograr un servicio fiable y acorde a las exigencias de los usuarios y conductores del transporte urbano selectivo y a la vez siguiendo lo reglamentado en la Ley 524: Ley General de Transporte Terrestre.

2 OBJETIVOS

2.1 Objetivo General.

Diseñar y construir un taxímetro digital con la capacidad de facturar cuatro carreras distintas de manera simultánea utilizando la tecnología GPS para una mejor regulación del transporte urbano selectivo.

2.2 Objetivos Específicos.

Identificar los principales requerimientos y limitantes para el diseño del taxímetro digital multipasajeros.

Realizar el análisis económico relacionado al proyecto.

Diseñar el circuito del taxímetro y visualizarlo mediante el uso de una herramienta informática competente y conveniente que permita su simulación y construcción del PCB correspondiente.

Adaptar un módulo GPS y un puerto para conectar una impresora de recibos al diseño propuesto.

Desarrollar la programación del microcontrolador a usar mediante una herramienta informática adecuada y accesible.

Ensamblar el dispositivo de manera que ocupe el menor espacio posible dentro del vehículo.

3 JUSTIFICACIÓN

En Managua aproximadamente 1,000,000 de personas hace uso del servicio de transporte público diariamente, de los cuales el 30% es brindado por el sector de transporte urbano selectivo. El transporte urbano selectivo a diferencia del transporte urbano colectivo no cuenta con una regulación económica, podemos afirmar que en Managua diariamente aproximadamente 300,000 personas pagan por un SERVICIO PUBLICO un precio sin fundamentos técnicos, sino basado en lo que el prestador del servicio considere justo. Es importante recalcar que la falta de regulación afecta ambas partes, ya que el taxista no considera todos los activos, fijos y variables, que incurre al momento de brindar el servicio.

Por otro lado el Instituto Regulador de Transporte del Municipio de Managua ha otorgado 10,500 concesiones, sin embargo por la capital circulan una cantidad mayor a ésta. Para el usuario es casi imposible conocer si el taxi que aborda está habilitado para brindar el servicio, hecho que atenta contra la seguridad ciudadana. La seguridad ciudadana y la justicia económica al abordar un taxi, son los motivos por los cuales se procedió a desarrollar este proyecto tecnológico, aplicando los conocimientos adquiridos durante el desarrollo universitario para dar una solución a esta problemática y brindar un aporte a la sociedad.

Esta propuesta se presenta en forma de un dispositivo electrónico llamado taxímetro digital multipasajeros que, a diferencia del taxímetro común, puede realizar hasta cuatro carreras distintas de manera simultánea, siempre y cuando las carreras sean sobre la misma ruta. Esto basado en la Ley 524, aprobada el 13 de Junio de 2005, Capítulo XVI, Arto. 99 que dicta que “La tarifa (de las carreras realizadas por los taxis) será establecida a través de un taxímetro, o en su ausencia por un contrato verbal entre el taxista y el usuario”, lo que indica que según la Ley, el taxímetro debería en teoría ser la principal forma de tasación de carreras de taxis en nuestro país.

4 MARCO TEÓRICO

4.1 Regulación del Sector Taxi en Managua

En Nicaragua la primera forma de regulación del sector taxi de la que se tiene documentación fue la división en zonas del departamento de Managua donde cada zona estaba identificada por una letra, de la A hasta la O, siendo así un total de 16 zonas, según el Acuerdo Ministerial No. 108, aprobado el 7 de Agosto de 1984 y publicado en El Diario Barricada edición No. 1766 del 12 de Agosto de 1984, iniciativa de ley con nombre: Tarifa Nocturna de Taxis Ruleteros de la Ciudad de Managua.³

Este sistema de tarifas convenía según el Arto. 1 en: Autorizar en VEINTE CORDOBAS la tarifa por pasajero transportado en Taxi Ruletero dentro del perímetro de una misma zona de la ciudad de Managua, incrementándose en DIEZ CORDOBAS el pasaje por cada cruce de zona en el traslado del pasajero a su destino.

Al día de hoy esto no tiene vigencia ya que la ley tocante al transporte es la Ley 524: Ley General de Transporte Terrestre, aprobada el 02 de Marzo de 2005 y publicada en La Gaceta No. 72 del 14 de Abril de 2005 y ampliada bajo el Decreto No. 42-2005, aprobado el 13 de Junio de 2005 y publicado en La Gaceta No. 113 del 16 de Junio de 2005, es la que regula el sistema de tarifación actual.⁴

Según el Arto. 2, inciso 1 de la Ley 524 aprobada el 2 de Marzo de 2005 los usuarios del transporte público tienen derecho a “Que se les cobre una tarifa

³ Acuerdo Ministerial No. 108; Tarifa Nocturna de Taxis Ruleteros de la Ciudad de Managua; Managua, Nicaragua, 07 de Agosto de 1984.

⁴ Ley 524: Ley General de Transporte Terrestre; Managua, Nicaragua, 16 de Junio de 2005.

acorde con la calidad del servicio que reciben y que la misma sea aprobada por autoridad competente”.

Según la Ley 524 ampliada, aprobada el 13 de Junio de 2005, en el Capítulo XVI, Arto. 95, el servicio de taxis locales es un servicio selectivo que utiliza un determinado número de personas sin rutas previamente establecidas... Su radio de acción geográfica estará circunscrito a los límites del casco urbano de la ciudad para la cual ha sido autorizado. El servicio de transporte selectivo se clasifica en:

- Taxis locales.
- Taxis de parada.
- Moto Taxi.

Los vehículos utilizados para brindar el servicio taxi debe cumplir con la rotulación adecuada que según la Ley 524 ampliada, Capítulo XVI, Arto. 97 es a como sigue: La rotulación en las puertas delanteras, deberá incluir como mínimo; el tipo de servicio, municipio donde opera, número de placas y organización a la que pertenece, y en la parte posterior deberá reflejarse únicamente el número de placas. En cada municipio se implementará el diseño de una raya de colores a lo largo de las partes medias laterales de los vehículos, y en el caso de que se opere por turnos, se establecerá por combinaciones diferentes para cada turno (blanco con negro y blanco con amarillo para los 2 distintos turnos en Managua, de 6AM a 2PM el turno matutino y de 2PM a 10PM el turno vespertino).

Cuando vehículos que circulan con rayas blanco con negro operan en el turno matutino, los vehículos que circulan con rayas blanco con amarillo deben operar en el turno vespertino y a la siguiente semana rotar los turnos; es decir, vehículos que circulan con rayas blanco con negro operaran en el turno vespertino la siguiente semana, y los vehículos que circulan con rayas blanco con amarillo deben operar en el turno matutino, y así consecutivamente. De

10PM a 6AM del día siguiente pueden operar vehículos con ambas combinaciones, así como el día domingo, pero IRTRAMMA solamente se responsabiliza por los trabajos realizados dentro de los turnos oficiales.

El trayecto del viaje o “carrera” se inicia en el momento del abordaje y termina en el lugar que solicite el usuario, siempre que dicho lugar sea accesible, según la Ley 524 ampliada, Capítulo XVI, Arto. 98.

El conductor priorizará el destino de los primeros pasajeros que abordaron el taxi, aunque en el trayecto podrá subir otros pasajeros SIEMPRE Y CUANDO VAYAN EN LA MISMA RUTA DE LOS PRIMEROS USUARIOS.

Por último, LA TARIFA SERÁ ESTABLECIDA A TRAVÉS DE UN TAXÍMETRO, o en su ausencia por un contrato verbal entre el taxista y el usuario, según la Ley 524 ampliada, Capítulo XVI, Arto. 99.

4.2 Qué es un Taxímetro⁵

Aparato contador que mide de forma continua el trayecto recorrido y establece el precio a pagar, según unas tarifas preestablecidas, por una determinada distancia. Se emplea en los automóviles públicos destinados al transporte de pasajeros y, en general, puede conectarse al cuentakilómetros. En caso de detenciones prolongadas, los saltos de taxímetro se suceden en relación directa con el tiempo transcurrido mediante un dispositivo temporizador.

⁵ m.diccionario.motorgiga.com/diccionario/taxímetro-definicion-significado/gmx-niv15-con195692.htm



Ilustración 1- Taxímetro Común

La invención del taxímetro no es reciente. Se tienen noticias seguras de su existencia ya en tiempos del emperador romano Marco Aurelio Cómodo, a finales del siglo II. Según algunas fuentes, la invención del taxímetro es muy anterior, remontándose a los tiempos de la gran civilización china; con ellos iban dotados unos vehículos públicos cuyo nombre podría traducirse como «coche con tambor que cuenta las distancias». Estos vehículos estaban formados por un solo timón, dos pequeñas ruedas y una caja con dos pisos. En cada uno de ambos compartimientos se hallaba situado un autómeta de madera que, con su mano derecha, sostenía un martillo; el del piso inferior daba un golpe sobre un tambor, gracias a un mecanismo accionado por las ruedas, cada vez que éstas recorrían una distancia de unos 1.600 m; en cambio, el del piso superior daba un golpe a un gong cuando la distancia recorrida era 10 veces mayor. Con este primitivo sistema, contando los golpes dados, era posible calcular la distancia recorrida y el precio a pagar.

4.2.1 Consideraciones en un Taxímetro⁶

El funcionamiento de un taxímetro depende de muchas consideraciones que dan la dirección hacia la cual debe moverse el diseño del mismo y este diseño debe

⁶ Guía de cuestiones a tener presentes en la definición de tarifas en el sector del taxi, Centro Español de Metrología, http://www.cem.es/sites/default/files/files/Guia_Tarifas_Taxi.pdf

asegurar una tarifa confiable. Las principales pautas bajo las cuales ha de regirse la tarifa o precio final son: un importe inicial, un importe kilométrico, el horario de aplicación de la tarifa y el calendario de días festivos.

- **Importe inicial:** es la tarifa base o mínima a pagar preestablecida por el organismo de regulación. Este importe es proporcional a la distancia de referencia, luego de la cual la tarifa aumentará conforme la distancia.
- **Importe kilométrico:** es el aumento en la tarifa a medida que aumenta la distancia una vez que se ha alcanzado la distancia de referencia.
- **Horario de aplicación de la tarifa:** el precio varía según el horario en que se realice la carrera, donde habrá un pequeño saldo extra a favor del chofer en horario nocturno.
- **Calendario de días festivos:** ligero aumento de la tarifa en días nacionalmente constituidos como feriados.

Otro indicador de gran peso que debe tomarse en cuenta para brindar la tarifa a pagar es el estado del vehículo, que se refiere al consumo de combustible, aceites, filtros, lavados, engrases, líquidos, baterías, correas, reparaciones mecánicas en general, tapizado, llantas, neumáticos y depreciación del vehículo.

También es necesario proveer al vehículo con el que se presta el servicio de un sistema de medición en perfecto estado de funcionamiento, que le permita al usuario saber con exactitud la suma por pagar según la distancia recorrida.

Por último, debe tomarse en cuenta el salario establecido por las autoridades correspondientes para el cadete del vehículo.

4.2.2 Funcionamiento de un Taxímetro

La manera en que el taxímetro está estructurado da una clara visión de su funcionamiento. De forma general se puede segmentar en tres partes: sensor de

distancia, circuito acondicionador, display (e impresora de recibos adicionalmente). El sensor de distancia envía una señal hacia el circuito electrónico que se encarga de procesar la información recibida para poder visualizarla a través del display. Generalmente el sensor de distancia es el odómetro del vehículo (aunque hoy en día también esto puede realizarse a través de GPS), y el display es una pantalla LCD (más adelante se hablará a detalle de esto).

4.2.3 El Taxímetro y sus Partes

Las partes que integran el taxímetro varían en dependencia de su fabricante pero la presentación usual del taxímetro consta de un sensor de medición de distancia (generalmente un cable conectado al odómetro o cuentakilómetros del vehículo. Posiblemente también a través de GPS), un circuito electrónico con un microcontrolador que se encarga de procesar la señal según la programación que se le haya compilado, una pantalla LCD que muestra la tarifa en todo momento, una impresora de recibos como módulo adicional adaptable, y botones de mando para indicar el inicio y fin de carrera.

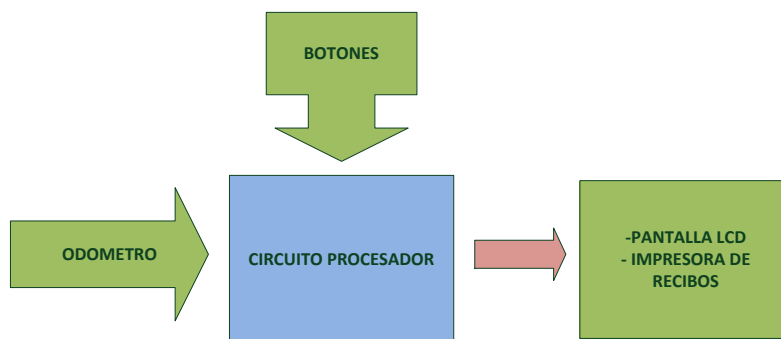


Ilustración 2- Diagrama en Bloques del Taxímetro Común

4.3 GPS (Global Positioning System)⁷

El Sistema de Posicionamiento Global (GPS) es un servicio propiedad de los EE.UU. que proporciona a los usuarios información sobre posicionamiento, navegación y cronometría. Este sistema está constituido por tres segmentos: el segmento espacial, el segmento de control y el segmento del usuario. La Fuerza Aérea de los Estados Unidos desarrolla, mantiene y opera los segmentos espacial y de control.

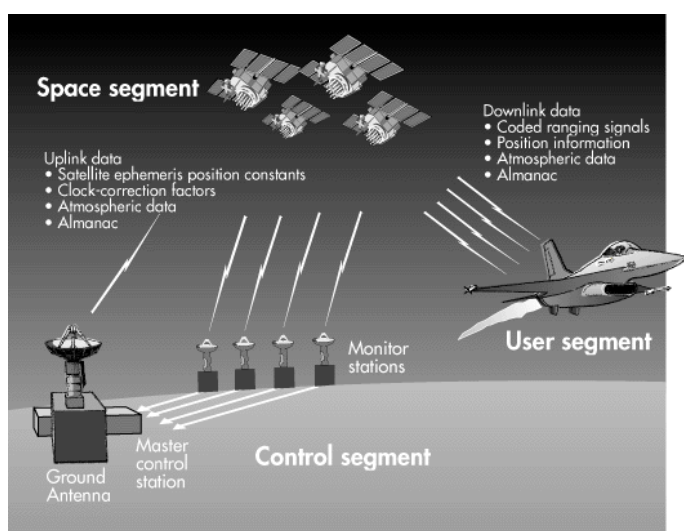


Ilustración 3- Funcionamiento GPS.

4.3.1 Segmento Espacial:

El segmento espacial GPS consiste en una constelación de satélites que transmiten señales de radio a los usuarios. Los Estados Unidos se comprometen a mantener la disponibilidad de al menos 24 satélites GPS operacionales el 95 % del tiempo. Para garantizar este compromiso, la Fuerza Aérea ha estado volando 31 satélites GPS operacionales en los últimos años.

Los satélites GPS orbitan el planeta tierra a una altitud de aproximadamente 20.200 kilómetros. Cada satélite rodea la Tierra dos veces al día. Los satélites

⁷ <http://www.gps.gov/systems/gps/>

de la constelación están dispuestos en seis planos orbitales igualmente espaciados en torno a la Tierra, esta configuración permite que los usuarios pueden ver al menos cuatro satélites desde prácticamente cualquier punto del planeta.

Los satélites transmiten dos señales de radio de baja potencia denominadas L1 y L2 y se derivan de una frecuencia común $f_0 = 10.23 \text{ MHz}$:

$$f_{L1} = 154f_0 = 1575.42 \text{ MHz}$$

$$f_{L2} = 120f_0 = 1227.60 \text{ MHz}$$

Las señales están compuestas de tres partes:

- Portadora: las dos opciones fundamentales actualmente son L1 y L2.
- Datos de navegación: contienen información acerca de las órbitas satelitales y la información de reloj; esta información se envía a todos los satélites desde las estaciones del segmento de tierra del GPS. La tasa de bits de los datos de navegación es de 50 bps.
- Secuencia de ensanchado: como ocurre en el caso de las comunicaciones móviles, la señal GPS contiene un código o secuencia que ensancha el espectro más allá de lo que le correspondería de acuerdo a la información que transporta; este código tiene la forma de un pseudoruido, que modula la portadora de tal forma que aumente el ancho de banda de la transmisión y reduzca la densidad de potencia espectral (es decir, el nivel de potencia en cualquier frecuencia dada); la señal resultante tiene un espectro muy parecido al del ruido, de tal forma que a todos los radiorreceptores les parecería ruido menos al que va dirigida la señal; este tipo de modulación se conoce como DS-CDMA (acceso

múltiple por división de código en secuencia directa); cada satélite utiliza dos de estos códigos:

- Código de Adquisición Aproximativa o C/A (course acquisition code) que da lugar al servicio estándar civil SPS (standard positioning service): se trata de una secuencia de 1023 chips (un chip es equivalente al concepto de un bit, pero recibe este nombre porque no pertenece a una palabra o byte de información sino a un código de identificación); este código se repite cada milisegundo, lo que da lugar a una tasa de chipping de 1.023 MHz; este código modula únicamente la señal L1 y es diferente para cada satélite.
- Código de Precisión (P(Y)), que se encuentra encriptado y permite el servicio para fines militares y de seguridad PSP (precision positioning service); es un código de mayor longitud, de unos 2.35×10^{14} chips, con una tasa de chips de 10.23 MHz (es decir, los chips son diez veces más cortos que para el C/A); este código, en lugar de repetirse cada milisegundo como el C/A, se repetiría, siguiendo una tasa de 10.23 MHz cada 266.4 días pero se realiza un reset a las cero horas del domingo y cada satélite utiliza un fragmento diferente para que se pueda producir la identificación del mismo a través del código (así podemos llegar a tener 38 satélites en funcionamiento simultáneo ($266.4/7 \approx 38.06$); el código P(Y) es militar y está encriptado, como hemos dicho, y modula tanto la L1 como la L2.

4.3.2 Segmento de Control:

El segmento de control de GPS consta de una red global de instalaciones de tierra que hacen un seguimiento de los satélites GPS, monitorean sus transmisiones, realizan análisis, y envían comandos y datos a la constelación.

El segmento de control de la corriente de trabajo incluye una estación de control principal MCS, una estación de control maestro alternativo, 11 antenas de mando y control, y 15 estaciones de monitoreo.

La estación de control principal MCS (Master Control Station) es la que realiza las funciones de control de segmentos primarios, proporcionando el mando y control de la constelación GPS. El MCS recibe información de navegación desde las estaciones de monitoreo y utiliza esta información para calcular la ubicación exacta de los satélites GPS en el espacio y verificar el estado de los relojes. Además, tiene capacidad para enviar correcciones de reloj, ajuste de órbita de los satélites, y enviar otros mensajes a través de las antenas en tierra.

4.3.3 Segmento Usuario:

El segmento del usuario consiste en el equipo receptor del GPS que recibe las señales de los satélites del GPS y las procesa para calcular la posición tridimensional y la hora precisa.

Un receptor GPS está compuesto de una antena, un sintonizador de frecuencias, un procesador y un reloj muy estable. Un receptor es generalmente descrito por el número canales, es decir, el número de satélites que se pueden monitorear simultáneamente.

Los receptores GPS pueden dividirse según el tipo de nivel de usuario en: los que usan SPS (Standard Positioning Service) Servicio de Posicionamiento Estándar que tienen una precisión aproximada de 7.8 metros el 95% del tiempo y los que utilizan PPS (Precise Positioning Service) Servicio de Posicionamiento Preciso. El PPS está reservado para uso militar y el SPS para uso civil o científico.

4.3.4 Funcionamiento del GPS⁸

El sistema GPS determina las coordenadas espaciales de un objeto a partir de sus distancias a otros objetos (satélites) cuyas posiciones son conocidas. La distancia entre un objeto y cada satélite se mide en términos del tiempo de propagación de las señales desde el satélite hasta dicho objeto. Para conseguir suficiente precisión es indispensable disponer, por un lado, de relojes ultra estables y, por otro, de plataformas espaciales estables localizadas en órbitas predecibles.

El GPS depende en que cada satélite en la constelación transmita su posición exacta y una señal de tiempo extremadamente precisa a los receptores en la tierra. Dada esta información, los receptores GPS pueden calcular su distancia al satélite, y combinando esta información de cuatro satélites, el receptor puede calcular su posición exacta usando un proceso llamado trilateración.

Si se conoce la distancia a un satélite, se sabe que su posición se encuentra sobre una esfera con centro en el satélite y con un radio igual a la distancia. Si se obtiene la misma información de un segundo satélite, puede estrechar su posible posición al área que tienen en común las dos esferas. Si se añade información de un tercer satélite, se puede precisar aún más la posición a los dos puntos donde las tres esferas cruzan. Este es el proceso conocido como trilateración.

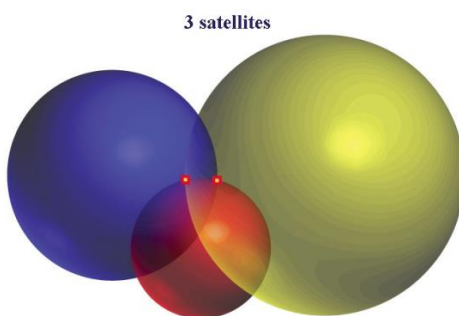


Ilustración 4- Representación del Proceso de Trilateración

⁸ <http://edis.ifas.ufl.edu/pdf/IN/IN65700.pdf>

Para determinar cuál de los dos puntos representa la posición actual, se puede tomar una cuarta medida, pero generalmente uno de los dos puntos obtenidos de tres satélites representa una posición absurda (por ejemplo en el espacio abierto) o con movimiento imposiblemente rápido, por lo cual se puede eliminar sin necesidad de la cuarta medida. Sin embargo, la cuarta medida, o cuarto satélite, aún es necesaria.

La cuarta esfera que es proyectada por el cuarto satélite debe cruzar las otras tres en un punto que representa la posición actual. Si existen errores, la cuarta esfera no cruzará a todas las otras. Debido a que el error del receptor es el mismo para las cuatro medidas, un ordenador en el receptor puede calcular una corrección que haga que las cuatro esferas crucen, y aplicar la corrección a las medidas para obtener la posición correcta.

A bordo de cada satélite se transportan relojes que generan frecuencias de referencia de gran exactitud (relojes atómicos de cesio o rubidio), a partir de las cuales se calculan las referencias temporales. En cuanto a los terminales de usuario, se tolera que estos incorporen relojes de cuarzo, menos precisos pero de menor coste económico.

Ahora bien, la correcta medida de los tiempos de propagación exige que los relojes de los satélites y de los receptores estén sincronizados. Debido a la menor calidad del reloj del terminal, es probable que en el momento de la medida exista un error entre los tiempos de tránsitos reales y los estimados en el receptor. Por este motivo, un receptor precisa tener a la vista un número mínimo de cuatro satélites para poder estimar sus tres coordenadas espaciales más la coordenada temporal.

La cuarta medida también permite al sistema calcular una posición en tres dimensiones que incluye no solo latitud y longitud, sino también altitud. Las medidas de altitud, sin embargo, se reportan con referencia a un indicador

convencional (normalmente nivel del mar). El modelo es solo una aproximación por lo cual las medidas de altitud son menos precisas que las de latitud y longitud, y los errores son diferentes en distintas partes del planeta.

4.3.5 Protocolo NMEA 0183⁹

Uno de los protocolos de transmisión de datos más utilizado para señales GPS es el NMEA 0183 que consiste en un protocolo que define los requerimientos de datos y tiempo de transmisión en el formato serial a una velocidad de 4800 baudios.

Los mensajes enviados bajo el protocolo NMEA son variados y pueden ser:

- GGA: Tiempo, posición y datos fijos.
- GLL: Latitud, longitud, tiempo UTC de fijación de datos y estado.
- GSA: Modo de operación del receptor GPS, satélites usados en el posicionamiento, y valores DOP (dilución de la precisión).
- GSV: El número de satélites GPS a la vista, ID de los satélites, elevación, azimut, y valores SNR.
- MSS: Proporción de señal a ruido, fuerza de la señal, frecuencia, y tasa de bits desde un radiofaro receptor.
- RMC: Tiempo, fecha, posición, curso y velocidad de los datos.
- VTG: Curso y velocidad de la información relativa al suelo.
- ZDA: Mensaje de sincronización PPS.
- 150: OK para el envío de mensajes.

El protocolo NMEA 0183 se basa en cadenas. Cada cadena inicia con un signo \$ (código ASCII 36), el significado de toda la cadena depende de la primera palabra. Por ejemplo, una cadena que comienza con \$GPGLL da información

⁹ NMEA: Manual de Referencia, disponible en:
<https://www.sparkfun.com/datasheets/GPS/NMEA%20Reference%20Manual1.pdf>

acerca de la latitud y longitud, hora exacta, validez de los datos (A-Activo o V-Prohibido) y la suma de verificación (“checksum”) que nos permite comprobar si los datos se recibieron correctamente. Los datos individuales de cada cadena están separados por coma “,”.

A continuación se muestra un ejemplo de una cadena NMEA completa:



Ilustración 5- Trama NMEA 0183

4.3.6 Sistema de Coordenadas Geográficas

Las coordenadas geográficas son una forma de designar un punto sobre la superficie terrestre utilizando dos coordenadas angulares, latitud (Norte y Sur) y longitud (Este y Oeste) expresadas en grados sexagesimales.

La latitud de un punto determina el ángulo formado por la vertical a la tierra que pasa por dicho punto con respecto al plano Ecuador. Las líneas de latitud se denominan paralelos y son las líneas de intersección de los infinitos planos perpendiculares al eje terrestre con la superficie de la tierra. La latitud máxima o mínima va desde los 0° hasta los 90°, los que se encuentran al norte del Ecuador reciben la denominación Norte y los que se encuentran al sur del Ecuador reciben la denominación Sur. Al Ecuador le corresponde la latitud 0°.

La longitud mide el ángulo a lo largo del Ecuador desde cualquier punto de la Tierra, las líneas de longitud son círculos que pasan por los polos y se llaman meridianos. El sistema toma como origen para designar la posición un determinado meridiano, denominado meridiano 0°, cuyo nombre toma el de una ciudad inglesa por el que pasa; GREENWICH.

La designación de la longitud es designada con respecto al meridiano Greenwich, se denomina Oeste cuando está a la izquierda del meridiano de origen y Este cuando está situado a la derecha. La longitud máximo o mínimo va de 0° hasta 180°.

Nicaragua se encuentra ubicada geográficamente entre los paralelos 10°45 y 15°15 de latitud Norte y entre los meridianos 83°00 y 88°00 de longitud Oeste

4.4 Comunicación USB¹⁰

El USB es un bus punto a punto donde solamente hay un único host (generalmente el PC) y donde pueden haber hasta 127 periféricos. Utiliza cuatro líneas de las cuales dos son para alimentación (Vcc y GND) y las otras dos para la transmisión de datos (D+ y D-). El host controla la velocidad del bus USB y se pueden dividir en:

- Low speed: 1,5 Mbps. Soportado por las especificaciones 1.1, 2.0 y 3.0.
- Full speed: 12 Mbps. Soportado por USB 1.1, USB 2.0 y USB 3.0.
- High speed: 480 Mbps. Solo USB 2.0 y USB 3.0.
- Super speed: 5Gbps solo soportado en dispositivos USB 3.0.

El host es el encargado de detectar cualquier dispositivo que se conecta al bus. Cuando un dispositivo es detectado el Host necesita obtener información sobre

¹⁰ <http://www.aquihayapuntes.com/indice-practicas-pic-en-c/comunicacion-usb-pic18f4550-utilizando-la-clase-cdc.html>

él, a este proceso se le conoce como enumeración. Esta información que necesita el Host se encuentra definida en el dispositivo en los llamados descriptores que permiten comunicar la identidad de un periférico. Los descriptores son datos que se guardan en la memoria no volátil del dispositivo y contienen la siguiente información: El ID del vendedor (VID) y del producto (PID), consumo de corriente del dispositivo, tipo de transferencia que se va a utilizar, etc.

El VID y el PID son dos números de 16 bits representados en Hexadecimal, con combinación única que deben ir programados en el driver USB. Los códigos VID son vendidos a los diferentes fabricantes de hardware del mundo por la corporación internacional USB-IF (<http://www.usb.org>), encargada de la estandarización y especificación del protocolo USB, y los códigos PID son elegidos por cada fabricante por su cuenta.

4.4.1 Clases de USB

La especificación USB provee de propiedades y funciones que pueden ser utilizadas por los dispositivos que tengan características similares. Por ejemplo, un teclado y un ratón por sus características pertenecerán a la misma clase llamada Human Interface Device (HID), una ventaja de utilizar esta clase por ejemplo es que no se necesita instalar ningún driver para el dispositivo ya que el sistema operativo utilizará uno genérico para todos. Las clases más utilizadas con Microcontroladores son:

- HID (Human Interface Device): ejemplos de dispositivos que utilizan esta clase como hemos dicho ya son: teclados, ratones, pantallas táctiles, joystick, etc. Velocidad low-speed (64 KB/s de velocidad máxima), tipos de transferencias soportadas: de control y de Interrupción. Una característica interesante al utilizar esta clase es que no se necesita

instalar un driver específico en el Sistema Operativo, se utiliza uno estándar que ya está incluido en el sistema.

- MSD (Mass Storage Device Class): Como su propio nombre indica para dispositivos de almacenamiento masivo como discos duros, memorias flash, cámaras digitales, dispositivos ópticos externos como lectores y grabadoras de CD y DVD, etc. Esta clase se puede utilizar solo en dispositivos que soporten velocidades Full y High Speed. El tipo de transferencias utilizadas es Bulk o una combinación formada por transferencias del tipo Control, Bulk y Interrupt. No se necesita la instalación de un driver específico, se utilizan drivers genéricos instalados ya en los Sistemas Operativos, en Windows se utiliza el driver llamado usbstor.sys ubicado en C:\Windows\System32\drivers.
- CDC (Communications Device Class): Un ejemplo de dispositivo que utiliza esta clase son los Módems. La velocidad máxima al utilizar esta clase será de 80 kBytes/s y el tipo de transferencias soportadas son del tipo interrupción y Bulk. Utiliza también driver estándar incluidos ya en el sistema operativo.

4.5 El Microcontrolador (μ C)¹¹

El Microcontrolador es un circuito integrado de alta escala de integración de bajo consumo que actúa como una minicomputadora, es decir, que incorpora la mayor parte de los elementos que configuran una computadora, como memoria de programa, memoria de datos y puertos para comunicarse con el exterior.

¹¹ http://galia.fc.uaslp.mx/~cantocar/microcontroladores/SLIDES_8051_PDF/2_INTROD.PDF

El microcontrolador es un sistema cerrado que contiene un computador completo y de prestaciones limitadas que no se pueden modificar. Dispone normalmente de los siguientes componentes:

- Procesador o CPU (Unidad Central de Proceso).
- Memoria RAM para contener los datos.
- Memoria para el programa tipo ROM/PROM/EPROM.
- Líneas de E/S para comunicarse con el exterior.
- Diversos módulos para el control de periféricos (temporizadores, Puertos Serie y Paralelo, CAD: Convertidores Analógico/Digital, CDA: Convertidores Digital/Analógico, etc.).
- Generador de impulsos de reloj que sincronizan el funcionamiento de todo el sistema.

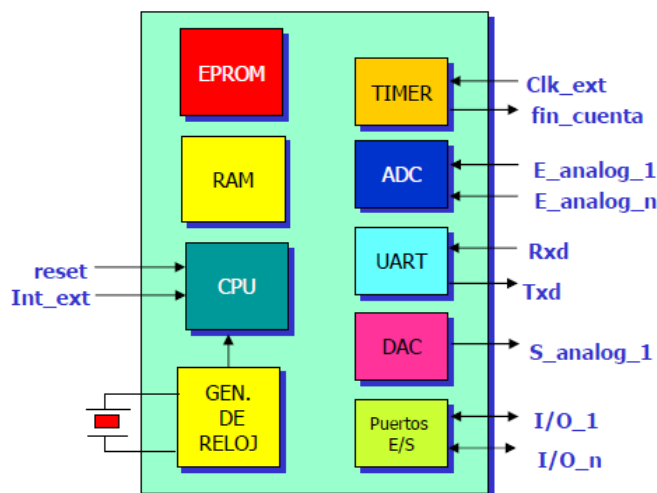


Ilustración 6- Componentes Comunes del Microcontrolador

Existen diferentes tipos de microcontroladores de diferentes capacidades; pueden ser de 4 bits, 8 bits, 32 bits, etc. Entre los MCU de 8 bits la arquitectura AVR es de los más populares debido a su sencillez y a que fue uno de los primeros en la historia en tener capacidad de almacenamiento y de reprogramación.

Los AVR utilizan la arquitectura Harvard, que separa el bus de datos y el bus de memoria. Las instrucciones en la memoria de programa son ejecutadas con una segmentación de 2 etapas. Mientras una instrucción está siendo ejecutada, la siguiente instrucción es pre-capturada de la memoria de programa (fetching).

4.5.1 Microcontrolador ATmega 328P¹²

El Atmega328P es un Microcontrolador CMOS de 8 bits basado en Arquitectura RISC de AVR. A continuación se presenta la descripción de los pines en el encapsulado DIP (Dual Inline Package).

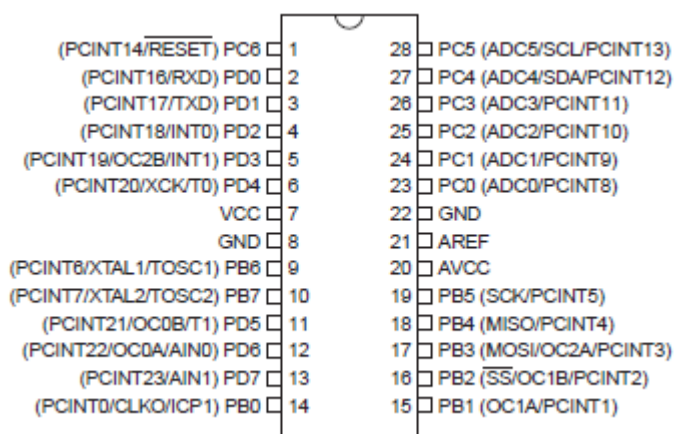


Ilustración 7- Atmega328P-PU.

El Atmega328P posee 131 instrucciones potentes ejecutadas en un solo ciclo de reloj y 32 registros de propósito general, alcanzando un rendimiento de 1 Millón de Instrucciones Por Segundo MIPS por MHZ, operable hasta 20MHz.

Las características principales del Atmega328P son:

- 32K Byte de memoria Flash programable.
- 1K Byte de memoria EEPROM.

¹² Datasheet ATmega328P

- 2K Byte de SRAM.
- Temporizadores y contadores.
- 8 canales A/D de 10 bits.
- 1 puerto serial SPI.
- 1 puerto I2C.
- Interrupciones internas y externas.
- 23 líneas I/O programables.
- Voltajes operables de 1.8 a 5.5 V.
- Rango de temperatura de -40 °C a 85 °C.

4.5.1.1 Memorias del Microcontrolador

4.5.1.1.1 Memoria Flash

El ATmega328P contiene una memoria Flash reprogramable de 32 KB para el almacenamiento del programa. Puede realizar al menos 10, 000 ciclos de escritura/borrado. La memoria Flash de programa está dividido en dos secciones: la sección del BootLoader y la sección de aplicación del programa.

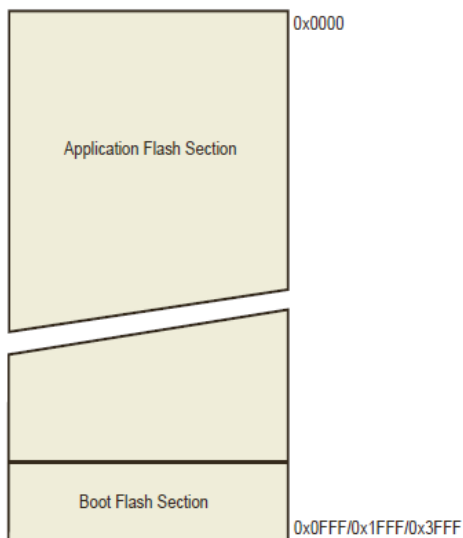


Ilustración 8- Memoria Flash

4.5.1.1.2 Memoria SRAM

El ATMega328P cuenta con una memoria SRAM de 2 KB. Esta memoria se divide en 4 secciones en las cuales se utilizan en la primera sección 32 espacios de memoria para registros de propósito general, en la segunda 64 espacios para registros de I/O, la tercera ocupa 160 espacios para registros de I/O extendidos, y el restante se utiliza para la memoria SRAM interna.

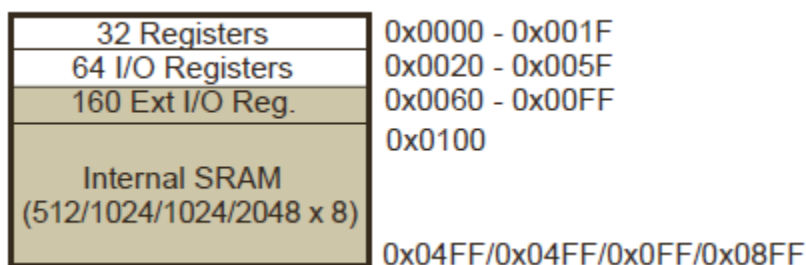


Ilustración 9- Memoria SRAM

4.5.1.1.3 Memoria EEPROM

El ATMega328P posee una memoria EEPROM de 1 KB. Está organizada como un espacio de memoria aparte, en el cual bytes particulares pueden ser leídos y escritos. Tiene una capacidad de 100,000 ciclos de escritura/borrado.

4.5.1.2 Líneas I/O Programables

Todos los puertos de los AVR tienen funcionalidad de lectura – modificación – escritura cuando son usados como puertos de I/O digital. Todos los puertos tienen resistores Pull-Up con resistencia invariante. Además poseen diodos de protección hacia Vcc y GND. Los puertos son representados por la nomenclatura PORTxn, donde x representa la letra de referencia para el puerto y n representa el número de bit; por ejemplo, PORTB3 indica el bit 3 para el puerto B.

Existen tres direcciones de memoria para las I/O en cada puerto, cada una para el registro de datos – PORTx, registro de dirección de datos – DDRx, y los pines de entrada del puerto – PINx. Las direcciones de las I/O de los pines de entrada de puerto (PINx) son únicamente de lectura, mientras que las de registro de datos (PORTx) y las de registro de dirección de datos (DDRx) son de lectura y escritura. El ATmega328P tiene 23 líneas que pueden ser programadas como entradas o salidas (I/O).

4.5.1.3 Interrupciones Externas e Internas:

Las interrupciones externas son disparadas por los pines INT0 e INT1 o por cualquiera de los pines PCINT23...0. Cabe destacar que las interrupciones se activarán aun si los pines INT0 e INT1 o los pines PCINT23...0 están configurados como salidas. Esta característica provee una forma de generar interrupciones por software. La interrupción por cambio en el pin PC12 se dispara si cualquiera de los pines PCINT23...16 habilitados cambia de estado. La interrupción por cambio en el pin PC11 se dispara si cualquiera de los pines PCINT14...8 habilitados cambia de estado. La interrupción por cambio en el pin PC10 se dispara si cualquiera de los pines PCINT7...0 habilitados cambia de estado. Los registradores PCMSK2, PCMSK1 y PCMSK0 controlan los pines que contribuyen a las interrupciones por cambio en los pines. Las interrupciones por cambio en los pines PCINT23...0 son detectadas de manera asíncrona. Esto implica que estas interrupciones pueden ser usadas para activar funciones desde el modo de reposo del MCU.

4.5.2 Microcontrolador ATmega8U2¹³

El ATmega8U2 es un MCU de 8 bits de bajo consumo con tecnología CMOS basado en la arquitectura RISC mejorada de AVR. Con la capacidad de ejecutar

¹³ Datasheet ATmega8U2

potentes instrucciones en un ciclo de reloj el ATmega8U2 alcanza un rendimiento aproximado de 1 MIPS por MHz permitiendo optimizar el consumo de energía y la velocidad de procesamiento. Las características principales del Atmega8U2 son:

- 8K Bytes de memoria Flash programable.
- 512 Bytes de memoria EEPROM.
- 512 Bytes de SRAM.
- Módulo USB 2.0.
- 1 puerto USART serie programable.
- 1 puerto serial SPI.
- 22 líneas I/O programables.
- Voltajes operables de 2.7 a 5.5 V.
- Rango de temperatura de -40 °C a 85 °C.

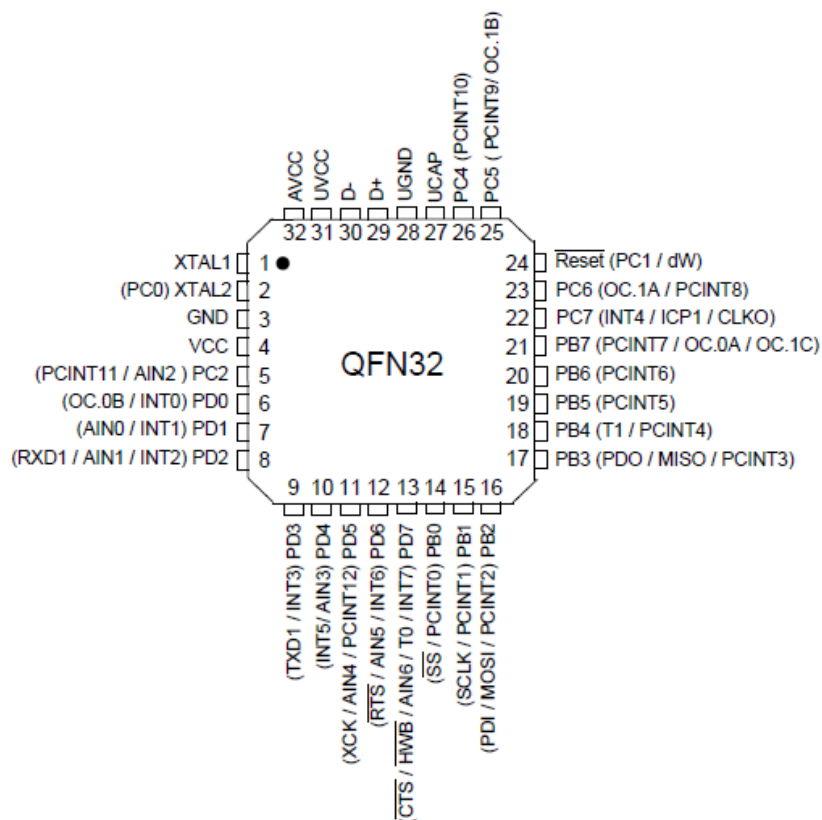


Ilustración 10- ATmega8U2-Au

4.5.2.1 Controlador USB:

El controlador USB implementa el hardware para una conexión USB 2.0 de máxima velocidad utilizando el ATmega8U2.

El controlador USB requiere un reloj de referencia de 48 MHz para la conexión. Este reloj es generado por un PLL interno. El reloj de referencia para el PLL debe ser proveído desde un cristal externo o desde una entrada de reloj externa. Solamente estas dos opciones de reloj serán capaces de proveer un reloj de referencia con la precisión y estabilidad requerida dentro de las especificaciones USB.

Para solventar las especificaciones eléctricas USB, los pines USB (D+ o D-) deben estar alimentados entre 3.0V y 3.6V. Como el ATmega8U2 puede ser alimentado hasta los 5.5V, un regulador interno provee el correcto voltaje a los pines USB. A continuación se muestra la implementación típica del Bus alimentada con 5V.

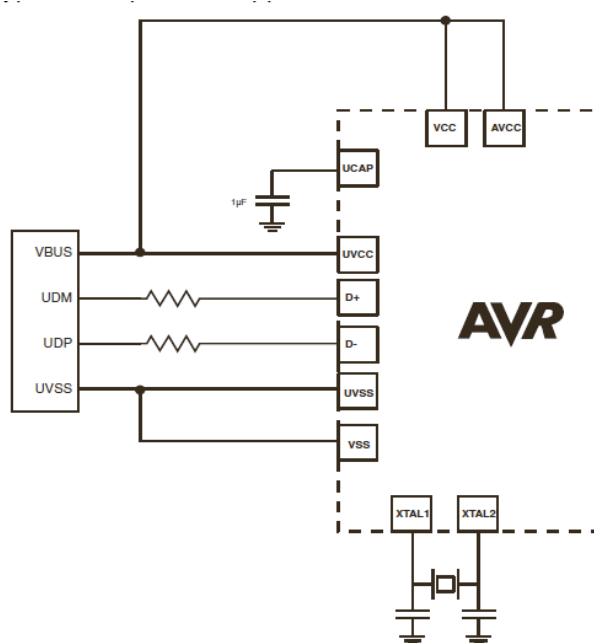


Ilustración 11- Conexión USB Atmega8u2.

5 DISEÑO DEL SISTEMA

El diseño del sistema está enfocado a cumplir con los requerimientos de funcionabilidad en base a lo establecido por la Ley General de Transporte Terrestre en el capítulo XVI, arto. 98-102. Pero debido a la falta de un modelo tarifario en Nicaragua, se procedió a realizar una propuesta, tomando como punto de partida datos oficiales que fueron facilitados por IRTRAMMA y la cooperativa FENICOOTAXI. Para ello nos hemos apoyado del Modelo de Regulación del Servicio Taxi utilizado en Costa Rica¹⁴.

5.1 Modelo Tarifario

Para la fijación de tarifas del servicio de taxi se deben contemplar todos los costos necesarios para prestar el servicio de manera que puedan cubrir básicamente 2 aspectos: todos los costos que son necesarios para garantizar el desarrollo de la actividad y una retribución competitiva al operador. Los costos totales se dividen en: Costos fijos y Costos variables.

5.1.1 Costos Fijos

Son aquellos que no varían con la cantidad de viajes realizados por el operador, como pagos administrativos, pagos de operación, pagos de salarios, entre otros.

COSTOS FIJOS		
RUBRO	FRECUENCIA	COSTO
3.1 PERMISO DE OPERACIÓN	Anual	C\$500.00
3.2 SEGURO RESP. CIVIL VEHICULO	Anual	C\$2,475.00

¹⁴ Modelo de Regulación Económica del Servicio Público de Transporte Remunerado de Personas, Modalidad Taxi.

3.3 SEGURO DE LICENCIA	Anual	C\$517.50
3.4 PAGOS A COOPERATIVA	Anual	C\$2,400.00
3.5 STICKER	Anual	C\$125.00
3.6 INSPECCION MECANICA	Anual	C\$250.00
3.7 FOTOS	Anual	C\$50.00
3.8 SALARIO	Anual	C\$120,000.00
3.8 AGUINALDO	Anual	C\$10,000.00
TOTAL		C\$136,317.50

Tabla 1- Costos Fijos en el Servicio Taxi

Estos datos, a excepción de los salarios y aguinaldos, fueron proporcionados por la cooperativa FENICOOTAXI y luego corroborados y actualizados acorde al mercado actual. Los salarios y aguinaldos fueron establecidos a criterio de los desarrolladores, quedando éstos anuentes a cualquier cambio. El total es la inversión anual en costos fijos por parte del taxista para brindar el servicio.

5.1.2 Costos Variables

Son los que varían en forma proporcional a la cantidad de kilómetros recorridos. Para fijar los costos variables se tomó como referencia un vehículo promedio de 1500cc.

COSTOS VARIABLES					
RUBRO	Unid. Medida	NORMA I	CANT/KMS	PRECIOS (C\$)	C\$/KM
1. GASOLINA SÚPER	Gln/Kms	1	48	C\$98.1178	2.0441
2. ACEITES					
Aceite No 40					
Base	Lts/Kms	1	6,000	C\$315.00	0.0525

Relleno	Lts/Kms	1	6,000	C\$88.00	0.0146
Aceite No 140					
Base	Lts/Kms	3	25,000	C\$97.00	0.0116
Relleno	Lts/Kms	0.5	25,000	C\$97.00	0.0019
3. FILTROS					
Filtro de aceite	Unid/Kms	1	6,000	C\$35.00	0.0058
Filtro de combustible	Unid/Kms	1	10,000	C\$86.00	0.0086
Filtro de aire	Unid/Kms	1	10,000	C\$90.00	0.009
4. LAVADOS					
Lavado de carrocería	Lav/Kms	1	240	C\$40.00	0.1666
Lavado de tapizados	Lav/Kms	1	5,000	C\$500.00	0.1
Lavado del motor	Lav/Kms	1	5,000	C\$140.00	0.028
5. ENGRASES					
Engrase general	Eng/Kms	1	6,000	C\$150.00	0.025
6. LIQUIDOS					
Líquido de frenos	Lts/Kms	1	20,000	C\$38.00	0.0019
7. BATERIAS Y CORREAS					
Baterías	Bat/Kms	1	108,000	C\$1,460.00	0.0135
Correas alternador	Corr/Kms	1	18,000	C\$65.00	0.0036
Correas hidráulica	Corr/Kms	1	18,000	C\$90.00	0.005
Banda de tiempo	Corr/Kms	1	80,000	C\$144.00	0.0018
8. MANT. TECNICOS MECANICOS	C\$/Kms	1	30,000	C\$6,000.00	0.2
9. REPARACIONES	C\$/Kms	1	25,000	C\$18,000.00	0.72

MECANICAS GENERALES					
10. TAPIZADO	Tap/Kms	1	125,000	C\$2,500.00	0.02
11. LLANTAS Y NEUMATICOS					
Llantas nuevas	Llant/Kms	5	43,200	C\$3,660.00	0.4236
Vulcanización	Vulc/Kms	1	1,000	C\$50.00	0.05
12. DEPRECIACION DEL VEHICULO	Unid/Kms	1	500,000	C\$175,000.00	0.35
TOTAL					4.2574

Tabla 2- Costos Variables en el Servicio Taxi

Al igual que los costos fijos, estos datos fueron proporcionados por la cooperativa de taxis FENICOOTAXI, actualizando los precios acorde al mercado actual.

El total de los costos variables es la inversión que realiza el taxista por cada kilómetro que recorre para brindar el servicio y éste varía según la distancia que recorre el operador por turno.

Se aplicará el modelo tarifario a los distintos tipos de vehículos, a efecto de calcular las siguientes tarifas:

5.1.3 Cálculo De La Tarifa

Para calcular la tarifa se requiere de facturar un componente por solicitar y recibir el servicio, que llamaremos tarifa inicial, y un componente por los kilómetros totales recorridos, que llamaremos tarifa variable por distancia.

La tarifa total será la suma de ambas tarifas, tarifa inicial y tarifa final. Para calcular cada una de ellas es necesario conocer los parámetros de servicio que el modelo de regulación económica requiere. A continuación se muestran los parámetros necesarios para calcular cada una de las tarifas.

5.1.3.1 Número De Días En Operación Por Año.

Para establecer este parámetro es necesario remitirse al Código del Trabajador tomando como premisa los 365 días del año.

Según el Arto. 76 de este código “Todo trabajador tiene derecho a disfrutar de quince días de descanso continuo y remunerado en concepto de vacaciones, por cada seis meses de trabajo ininterrumpido al servicio de un mismo empleador, por ende 30 días de vacaciones al año”.

Además hay que tomar en cuenta los días feriados oficiales, tales como:

Días Feriados	Motivo
01 de Enero	Año Nuevo
24 de Marzo	Jueves Santo
25 de Marzo	Viernes Santo
01 de Mayo	Día de los Trabajadores
19 de Julio	Aniversario de la Revolución Sandinista
14 de Septiembre	Aniversario de la Batalla de San Jacinto
15 de Septiembre	Día de la Independencia
08 de Diciembre	Día de la Concepción de María
25 de Diciembre	Navidad
Total: 9 Días Feriados Oficiales	

Tabla 3- Días Feriados Oficiales

También debe considerarse un día libre por semana, los séptimos días. Para 52 semanas en el año, 52 séptimos días.

Por tanto los días de operación en el año equivalen a 274 días como puede apreciarse en la siguiente tabla:

DIAS LABORALES EN EL AÑO	
DÍAS AL AÑO	365
SÉPTIMO DÍA	52
DÍAS FERIADOS	9
VACACIONES	30
TOTAL	274

Tabla 4- Días Laborales en el Año

5.1.3.2 Jornada Diaria del Taxista.

Los cadetes del servicio taxi trabajan en dos turnos de 8 horas cada uno: turno diurno, de 6:00 am a 2:00 PM, y turno nocturno, de 2:00pm a 10:00 PM. Si una semana un taxista trabaja en el turno diurno entonces a la siguiente semana trabajaría en el turno nocturno y así consecutivamente.

5.1.3.3 Número Promedio de Viajes por Día.

Según el último estudio de mercado realizado por IRTRAMMA en el año 2006 se realizan un promedio de 26.6 viajes diarios.

5.1.3.4 Kilómetros Diarios con Pasajeros

Según el último estudio de mercado realizado por IRTRAMMA en el año 2006 se recorren un promedio de 104.14 kilómetros diarios con pasajeros.

5.1.3.5 Kilómetros Totales Diarios

Según el último estudio de mercado realizado por IRTRAMMA en el año 2006 se recorren un promedio de 148.84 kilómetros diarios con un aprovechamiento del 70%. A continuación se muestran los parámetros necesarios para calcular la tarifa.

Parámetros	
Número de días en operación por año.	274 días
Jornada diaria del taxista.	8 horas
Número promedio de pasajeros por día.	26.6
Kilómetros diarios en que el taxi es utilizado	104.14 Km
Kilómetros totales diarios	148.84 Km
Aprovechamiento del recorrido	70%

Tabla 5- Detalle de Aprovechamiento Diario del Servicio Taxi

5.1.4 Ecuación del Cálculo de la Tarifa

5.1.4.1 Tarifa Inicial

La tarifa inicial está destinada a cubrir con los costos fijos que son invariantes a la cantidad de kilómetros que recorra el vehículo por día. Se calcula de la siguiente forma:

$$\text{Ec. (1) } TI = \frac{CFPA}{(NPPD) * (NDOA)}$$

Dónde:

- TI = Tarifa Inicial.
- CFPA = Costos Fijos Por Año.
- NPPD = Número Promedio de Pasajeros por Día.
- NDOA = Número de Días en Operación por Año.

Se calculó por año para tomar en cuenta todos los días que el trabajador del volante goza asueto de salario y para obtener una mayor precisión del costo.

5.1.4.2 Tarifa Variable por Distancia

La tarifa variable está destinada a recuperar la inversión que se realiza por brindar el servicio en unidades de kilómetros, que se expresó anteriormente (costos variables).

$$\text{Ec. (2) } TVPD = \frac{(CVPK) * (NPKRD)}{NPKRPD}$$

Dónde:

- TVPD = Tarifa Variable Por Distancia.
- CVPK = Costos Variables Por Kilómetro.
- NPKRD = Número Promedio de Kilómetros Recorridos por Día.
- NPKRPD = Número Promedio de Kilómetros Recorridos con Pasajeros por Día.

Es importante destacar que en la tarifa variable por distancia el taxista no tiene un margen de utilidad (sin ganancias) que evita que el conductor realice la carrera por la ruta más larga, ya que él no se beneficia de esto sino por el número de pasajeros que aborden la unidad.

5.1.4.3 Tarifa Total.

Por la particularidad del dispositivo, al momento de facturar más de una carrera, el precio de la tarifa variable por distancia se divide entre el número de pasajeros a bordo, el tiempo que compartan la unidad, es decir;

$$\text{Ec. (3)} \quad TT = TI + \frac{(TVPD * nK)}{n}$$

Dónde:

- TT = Tarifa Total.
- TI = Tarifa Inicial.
- TVPD = Tarifa Variable Por Distancia.
- n = número de pasajeros a bordo.
- nK = Kilómetros recorridos en la carrera.

Es decir que los costos en que se incurre por cada kilómetro recorrido se dividen entre la cantidad de pasajeros a bordo de la unidad el tiempo que éstos la comparten.

Para mayor comprensión se realiza el siguiente ejemplo: Calcular la tarifa total de 1 persona que se desplaza 5 Km de un punto A hacia un punto B.

$$TI = \frac{CFPA}{(NPPD) * (NDOA)} = \frac{C\$136,317.50}{(26.6) * (274)} = C\$18.70$$

$$TVPD = \frac{(CVPK) * (NPKRD)}{NPKRPD} = \frac{(C\$4.2574) * (148.84\text{Km})}{104.14 \text{ Km}} = C\$ 6.08$$

$$TT = TI + \frac{(TVPD * nK)}{n} = C\$18.70 + \frac{(C\$ 6.08) * (5 \text{ Km})}{1} = C\$ 49.12$$

5.1.4.4 Tarifa Extraordinaria

Por otro lado la tarifa para los días feriados y para los horarios que están fuera de los turnos regulares, es decir, de 10PM a 6AM, tienen un cargo extra siempre siguiendo lo estipulado en el Código del Trabajador en lo referente al trabajo en horas fuera del horario regular.

El cargo extra es un 50% sobre la tarifa total asegurando lo establecido, siendo la tarifa extraordinaria:

$$\text{Ec. (4) } TE = TT + \frac{TT}{2}$$

Dónde:

- TE = Tarifa Extraordinaria.
- TT = Tarifa Total.

5.2 Diseño del Prototipo

Las pretensiones del sistema son las de cumplir con la capacidad de realizar múltiples carreras, brindar tarifas fiables y un tener un tamaño de reducidas proporciones, entre las principales. Para este fin se ha construido un prototipo basado en el microcontrolador ATmega328P al cual se le ha incluido una pantalla LCD de 20x4, un módulo GPS, una impresora térmica de recibos, botones de mando y LEDs indicadores.

El sistema cuenta con 5 botones de mando que corresponden a: Nuevo Pasajero, Terminar Carrera, direccional hacia arriba, direccional hacia abajo y botón de intro/OK. El botón de Nuevo Pasajero permite iniciar una nueva

carrera. El botón de Terminar Carrera permite finalizar el trayecto recorrido. Los botones direccionales permiten moverse dentro de las opciones de los distintos menús que se muestren en pantalla. Y el botón intro/OK permite seleccionar la opción deseada en el menú respectivo.

Los LEDs indican el energizado del sistema, el estado de conexión del GPS, la comunicación USB, y la impresión en curso.

Adicionalmente se ha incluido un botón de RESET para propósitos meramente técnicos, no manejable por el usuario final.

El diseño del prototipo se ha seccionado en tres partes, que son:

- Diseño de hardware.
- Diseño de firmware.
- Diseño de empaquetado.

5.2.1 Diseño de Hardware

En esta sección se describe el manejo de entradas y salidas del microcontrolador ATmega328P, su intercomunicación con la computadora mediante el microcontrolador ATmega8U2, así como la selección de la interfaz visual, módulo GPS e impresora de recibos. También se aborda de manera breve el programa que se utilizó para el modelado 3D del circuito, Altium Designer Versión 13. A continuación se muestra el diagrama en bloques del desarrollo de hardware.

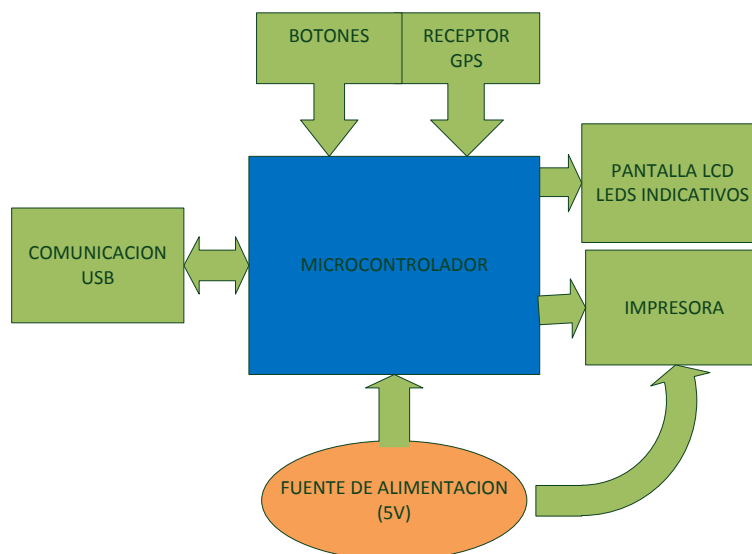


Ilustración 12- Desarrollo de Hardware

5.2.1.1 Altium Designer Versión 13

Altium Designer es una herramienta que permite el diseño profesional de PCBs y que combina el conocimiento científico con la intuición natural para crear una visión sencilla y unificada del circuito. Su estructura permite seguir líneas guías marcos precisos y ajustes exactos.

Altium Designer Versión 13.0 posee un área de trabajo (Workspace) que muestra una vista previa gráfica de todos los documentos contenidos en un proyecto abierto. Posee un editor de vértices de polígonos en PCB, regla de diseño de PCB, y la posibilidad de personalización en la designación de nombres de los materiales usados (posee una vasta librería de materiales), el tipo de letra y el color.

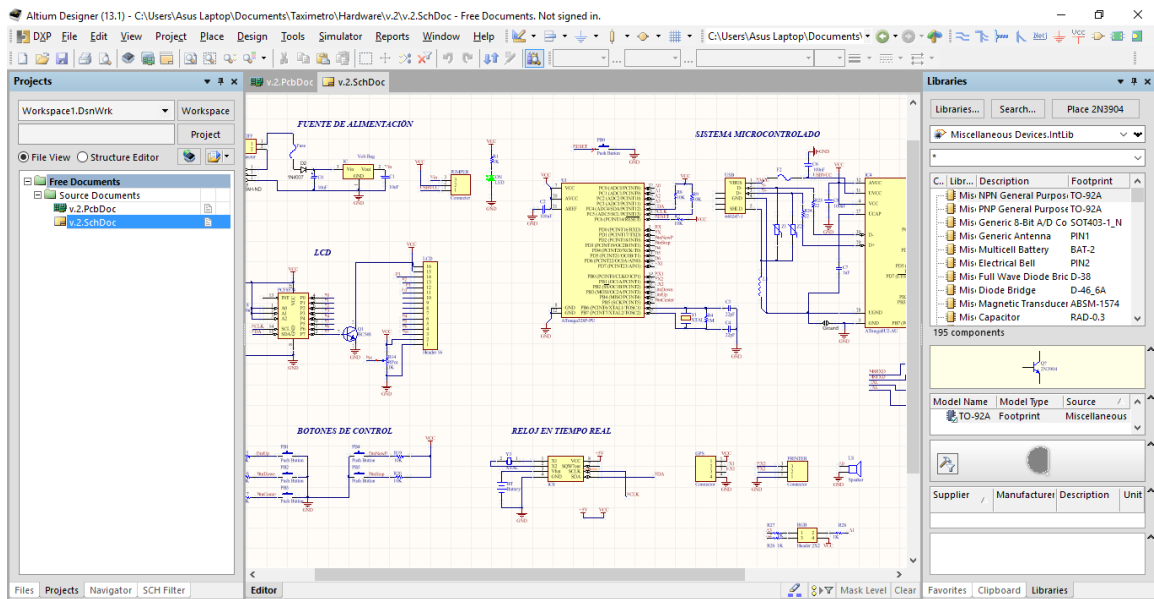


Ilustración 13- Editor de Esquemáticos Altium Designer

En el software de Altium, un proyecto de PCB es el conjunto de documentos de diseño (Diagrama esquemático, editor PCB, Explosión de materiales) que se requieren para especificar y fabricar una placa de circuito impreso. La forma de diseñar una PCB es similar a la de otros softwares, primero se requiere un diagrama esquemático con el diseño del circuito electrónico y se procede a simular el sistema permitiendo corregir errores. Cuando el diseño esquemático está libre de errores puede ser transferido al editor de trabajo PCB, utilizando un proceso conocido como la sincronización de diseño. La siguiente fase es la de diseñar el PCB de acuerdo con las reglas de diseño y por último se generan las salidas de fabricación y montaje.

Cada componente agregado en el diagrama esquemático tiene asociado un layout o footprint, lo cual permite la facilidad de manejo del editor de PCB al momento de exportar el diagrama esquemático. En la red existe una gran cantidad de librerías de componentes que pueden ser instaladas en el software.

5.2.1.2 ATmega328P

Como elemento principal de procesamiento se optó por el microcontrolador Atmega328P por la familiaridad de los desarrolladores con él, la amplia información que existe en la red, y su bajo coste que hacen de este una opción muy asequible. La distribución de los pines del microcontrolador se detalla de la siguiente manera:

- Puerto B configurado para la pantalla LCD. Utiliza los pines B5 al B2 del microcontrolador a los pines D7-D4 del LCD para transmisión de datos en paralelo, se conectó el pin B1 del microcontrolador al pin E del LCD y el pin B0 del microcontrolador al pin RS del LCD.
- Pin 6 y 7 del puerto B para el cristal de 16 MHz.
- Pines 0, 1 y 2 del puerto C como entradas de propósito general para los botones de direccionales: DOWN, UP e INTRO/OK, a través de resistencias de Pull-Up.
- Pines 3, 4, y 5 del puerto C como salidas de propósito general para el led RGB indicador.
- Pin 6 del puerto C para el botón de RESET.
- Puerto UART para la comunicación con el microcontrolador ATmega8U2 y con la impresora. Este corresponde a los pines 0 y 1 del puerto D.
- Pines de interrupción externa (INT0 e INT1) para el botón de nuevo pasajero y de terminar carrera. Éstos corresponden a los pines 2 y 3, respectivamente, del puerto D.

- Pin 4 del puerto D para el manejo del Buzzer.
- Pin 5 del puerto D para el manejo del backlight del LCD.
- Pin 6 y 7 del puerto D como I/O de propósito general para la comunicación con el GPS.

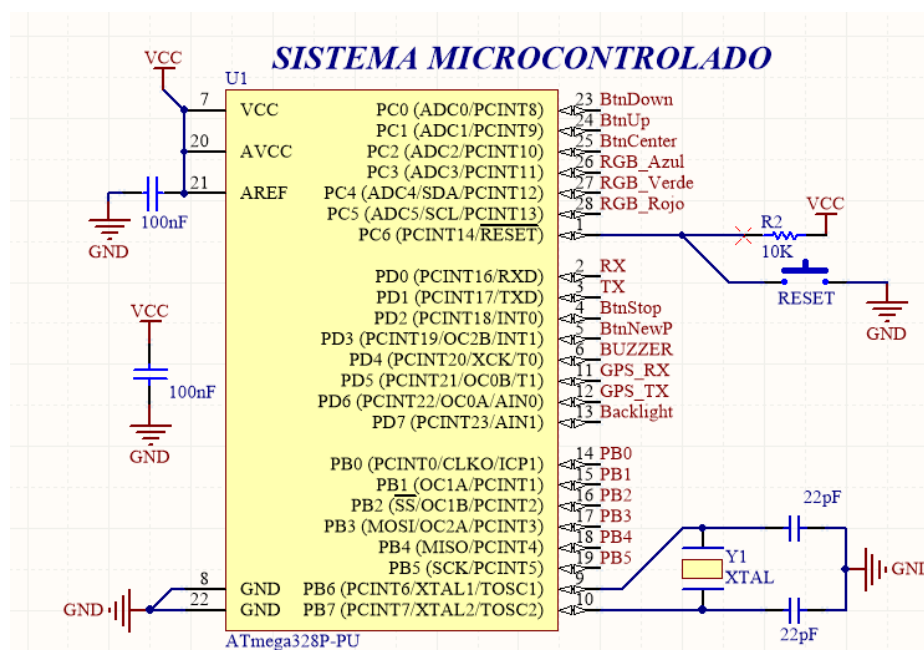


Ilustración 14- Distribución de Pines ATmega328P

5.2.1.3 Módulo GPS

El objetivo principal del módulo receptor GPS es obtener la posición, fecha, hora y la velocidad del automóvil en tiempo real para calcular por medio del mismo la distancia recorrida por el automotor. Para el caso en particular se requiere que la información solicitada se comuniquen con el microcontrolador mediante uno de los puertos seriales (UART TTL), con el fin de facilitar el manejo y procesamiento de estos datos.

Otro requerimiento importante del módulo GPS es su tamaño, para que pueda ser instalado en el interior del vehículo sin afectar la calidad de recepción de las tramas de datos. De igual manera se requiere que el receptor GPS opere a niveles de voltajes compatibles con el microcontrolador y que tenga un bajo consumo de energía ya que estará conectado a la batería del automóvil.

Según los requerimientos antes mencionados se procedió a investigar en el mercado y se escogió el módulo PMB-688 basada en el IC SiRFstar III. El módulo consiste en una placa de proporciones pequeñas con medidas de 33mm x 39mm que puede rastrear hasta 20 satélites.

El módulo GPS dispone de una antena integrada de cerámica que proporciona una sensibilidad de seguimiento de -159 dBm, además es posible conectar una antena externa a través del conector RF tipo MMCX para proveer mayor sensibilidad al módulo.



Ilustración 15- Módulo GPS

El módulo opera a 3.3 V a muy bajo consumo de energía durante la navegación (65mA), pero puede ser alimentada por los 5V que comúnmente utilizan los MCU ya que consta de un regulador de voltaje de 3.3V.

Sobre la placa hay una pequeña batería para proveer respaldo al GPS y una pequeña EEPROM conectada al GPS para guardar configuraciones permanentes. También sobre la placa hay un LED, que indica cuando el GPS está recibiendo datos. La placa tiene cuatro conectores principales: VCC, GND, TX (transmisión) y RX (recepción).

Este módulo tiene la capacidad de realizar un arranque en frío (cold start) en 42 segundos, arranque en caliente¹ (warm start) en 38 segundos, y un arranque en caliente² (hot start) en 1 segundo. Trabaja en la banda L1 correspondiente a los 1575.42 MHz, con un sistema de codificación C/A (Coarse/Acquisition). Además tiene una sensibilidad de seguimiento y navegación de -159 dBm y de readquisición de -148 dBm.

5.2.1.3.1 Interfaz de Comunicación

El módulo PMB-688 posee una interfaz serial TTL que opera a una velocidad de transmisión de 4800 baudios, con 8 bits, un bit de paro y sin bit de paridad. Los mensajes que transmite el módulo de manera asíncrona son: GSV, RMC, GSA, GGA, GLL, VTG, y TXT usando el protocolo NMEA 0183.

Para la comunicación del módulo con el microcontrolador se requieren de 2 pines de datos (RX y TX), donde el pin TX del receptor GPS se debe conectar al pin RX del microcontrolador y el pin TX del microcontrolador al pin RX del receptor GPS, más los pines de alimentación (5V y GND).

Es importante recalcar que se utilizaron los pines de propósito general 6 y 7 del puerto D del microcontrolador como puertos virtuales UART TTL, programados en el firmware.

5.2.1.4 Pantalla LCD

Uno de los requerimientos imprescindibles del taxímetro es la clara visualización de la información (Costo, tarifa, precio) por parte de los usuarios de hasta 4 carreras de manera individual, simultánea, y permanente mientras utilicen el servicio.

De igual manera se requiere que éste periférico se pueda comunicar con el microcontrolador utilizando la menor cantidad de pines posible, y por último y no menos importante, que el módulo sea de bajo consumo de energía.

Según los requerimientos antes mencionados se decidió utilizar el módulo LCD J204A de 20x4, 20 caracteres para cada una de las 4 filas que tiene disponible. Esta estructura permite que por cada fila de la pantalla se muestren los datos de interés por cada pasajero, es decir, la información de precios del pasajero 1 se mostraran en la fila 1 del LCD, los del pasajero 2 en la fila 2 del LCD y así sucesivamente.



Ilustración 16- Módulo LCD 20x4

Su bajo consumo de potencia, sus niveles de voltaje de operación que van desde los 3V hasta los 5V y su bajo costo lo hacen una opción muy viable para utilizarlo en el desarrollo del proyecto.

5.2.1.4.1 Características del Módulo LCD

Las pantallas de cristal líquido tienen la capacidad de mostrar cualquier caracter alfanumérico. La pantalla consta de 4 filas, cada una de ellas de 20 caracteres. Cada caracter consta de una matriz de 5x8 puntos, donde el proceso de

visualización es gobernado por un microcontrolador incorporado a la pantalla, siendo el Hitachi 44780 el modelo de controlador más utilizado.

➤ Descripción de los Pines del LCD:

- Pines 1 y 2: están destinados para conectar los 5V que requiere el módulo para su funcionamiento.
- Pin 3: es utilizado para ajustar el contraste de la pantalla; es decir colocar los caracteres más claros o más oscuros.
- Pin 4: el seleccionador de registro (RS) es el que permite sintonizar el modo de operación del bus de datos del LCD (modo de instrucción o modo caracter). Al colocar un 0 (Gnd) en el pin 4 se interpretará al bus de datos como una instrucción (por ejemplo: limpiar pantalla, mover cursor); y al colocar un 1 (5V) la información contenida en el bus de datos se interpretará como un caracter ("H", "\$").
- Pin 5: el pin R/W (leer/escribir) en 0 (Gnd) permite entrar en modo escritura y escribe en pantalla el dato presente en el bus de datos. Al colocar en 1 (5V) el pin R/W se lee el dato que está presente en el bus de datos.
- Pin 6: E (habilitación) es el que permite activar el reloj, de modo que el LCD puede ejecutar instrucciones que se den en ella, o en su defecto, inhabilitar las instrucciones omitiendo la información contenida en el bus de datos.
- Pines 7-14: denominado bus de datos, representa 8 líneas que se utilizan para colocar el dato que puede ser una instrucción o un caracter. En el bus de datos se pueden emplear 8 bits o solamente 4 bits. La diferencia entre uno y otro es el tiempo de acceso, pero en esta aplicación este

tiempo es despreciable por lo cual se procedió a utilizar bus de datos de 4 bits para optimizar el número de líneas a utilizar por el circuito electrónico.

- Pines 15 y 16: están destinados para suministrar la corriente al backlight o retro-iluminación.
-

En la siguiente tabla se muestra la distribución de los pines del módulo LCD.

■ PIN FUNCTIONS					
No	Symbol	Function	No	Symbol	Function
1	V _{SS}	GND, 0V	10	DB3	Data Bus
2	V _{DD}	+5V	11	DB4	—
3	V _{EE}	for LCD Drive	12	DB5	—
4	RS	Function Select	13	DB6	—
5	R/W	Read/Write	14	DB7	—
6	E	Enable Signal	15	LEDA	LED Power Supply
7-9	DB0-DB2	Data Bus Line	16	LEDA	

Tabla 6- Distribución de Pines del Módulo LCD

➤ Conexión del LCD al Microcontrolador ATmega328P

La conexión entre el microcontrolador y el LCD es a través del bus de datos y el bus de control (E, RS, R/W). El bus de datos del display es de 8 bits, pero se ha configurado para trabajar con cuatro, multiplexando los bytes en grupos de 4 bits (nibbles). La pantalla se ha configurado en modo escritura, por ello se utilizó el pin R/W en 0 conectado directamente a tierra. A continuación se muestra el diagrama de conexiones:

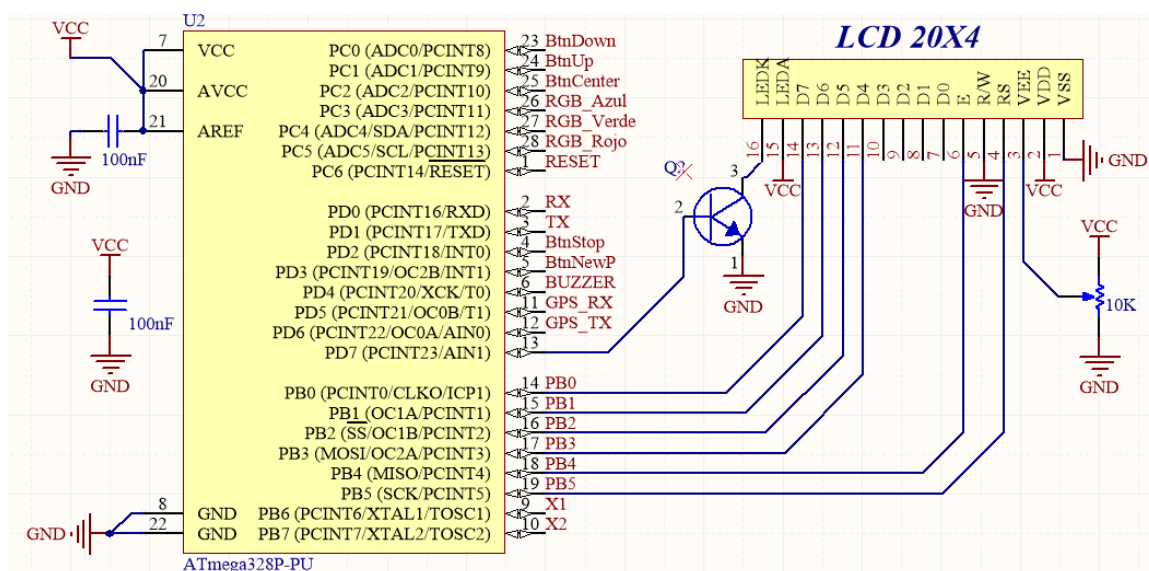


Ilustración 17- Conexiones LCD

Las conexiones son:

- Los pines d7-d4 del módulo LCD, se conectan a los pines 0, 1, 2 y 3 del puerto B del Atmega328p.
- El pin E del LCD al pin 4 del puerto B del Atmega328p.
- El pin R/W del LCD conectado directamente a tierra.
- El pin RS del LCD al pin 5 del puerto B del Atmega328p.
- El pin Vss del LCD a GND y Vdd del LCD a 5V.
- El pin VEE del LCD al potenciómetro para el ajuste del contraste.
- El pin LEDA del LCD a VCC y el pin LEDK del LCD al colector del transistor BC548 que actúa en corte y saturación conectado a la línea del pin 7 del puerto C del Atmega328p.

5.2.1.5 Selección de la Impresora

El objetivo principal de la impresora es la de emitir en principio un ticket donde le brinde al usuario el indicativo sobre cual pasajero es al abordar la unidad, y al final del recorrido emitir el recibo con la respectiva cuenta. Para la selección de la impresora se tomó como premisa que la impresora pueda ser instalada dentro del vehículo (tamaño reducido), que tenga un puerto TTL para la comunicación con el microcontrolador y que sea de bajo coste.

Se decidió optar por una mini impresora térmica CSN-A2, ya que posee buenas características tanto mecánicas como eléctricas que se adaptan perfectamente al prototipo.



Ilustración 18- Mini impresor CSN-A2

Esta impresora utiliza un cabezal térmico que es el encargado de reproducir la imagen o texto sobre el papel. El cabezal térmico cuenta con numerosos pines que se calientan a través de una resistencia, transmitiendo la imagen sobre el papel especial. Para lograr esta imagen, el cabezal térmico especifica qué pines deben calentarse y a qué temperatura deben hacerlo. El papel especial conocido como papel térmico es un papel que es reactivo al calor, cuando se aplica una fuente de calor se oscurece justo donde se aplica el calor.

A continuación se muestran las principales características de la mini impresora CSN-A2:

- La impresora tiene unas dimensiones de 111mm X 65mm X 57mm.
- Interfaz de comunicación TTL.
- Voltaje de alimentación de 5VDC a 9VDC
- Velocidad de impresión de 60 mm por segundo.
- Tiempo de vida de impresión de 50 Km.
- Rollo de papel 58mm de ancho.
- Opera en temperaturas desde 0 °C a 50 °C.

La impresora se conecta al puerto TTL del microcontrolador (pines 0 y 1 del puerto C) operando a una velocidad de 115200 baudios; por otra parte la impresora requiere al menos 1.5 A durante el período de impresión para funcionar correctamente.

5.2.1.6 Comunicación USB

En el taxímetro es necesario actualizar tarifas debido a los cambios que existen en los costos de los consumibles a través del tiempo y es por ello que el sistema debe tener la capacidad de conectarse a una computadora que le permita a la parte reguladora actualizar los precios en función a los cambios que sufren los costos de operación.

El microcontrolador principal Atmega328P transmite y recibe datos en niveles TTL (puerto UART); sin embargo la interfaz RS232 ha desaparecido de la nueva generación de computadoras y han sido reemplazadas por la interfaz USB, es por ello que es necesario migrar al interfaz USB.

- Pin 3 del puerto D (TX) conectado al pin 0 del puerto D (RX) del. Atmega328P a través de una resistencia de 1 K.
- Pin 4 del puerto D conectado al cátodo del LED indicativo RX.
- Pin 5 del puerto D conectado al cátodo del LED indicativo TX.
- D+ y D- conectados al D+ y D- del conector USB a través de resistencias de 22 Ohmios.
- UGND conectado V- del conector USB.
- UVCC conectado V+ del conector USB.
- UCAP conectado a una patilla de un capacitor de 1uF.

El Atmega8u2 está conectado al igual que la impresora al puerto TTL del microcontrolador Atmega328P, por ello es necesario desconectar la impresora al momento de utilizar la comunicación USB.

Otro aspecto importante es que podemos programar el microcontrolador Atmega328P directamente desde el puerto USB, sin la necesidad de usar herramientas de programación externas, siempre y cuando éste tenga un gestor de arranque (Bootloader) que abordaremos en secciones posteriores

5.2.1.7 Diseño de la Fuente de Alimentación

Para el diseño de la fuente de alimentación se consideró el consumo máximo de cada uno de los elementos que componen el sistema, teniendo en cuenta que el

sistema se conectará a los 12VDC procedentes de la batería del vehículo. A continuación se muestra la siguiente tabla con el consumo de potencia de cada elemento.

Dispositivo	Corriente máxima
Atmega328P	200 mA
Módulo GPS	67 mA
LCD	180 mA
LED	12 mA
Impresora	1500 mA
Total	1999 mA

Tabla 7- Consumo Detallado de Corriente del Dispositivo

Se excluye el Atmega8u2 debido que éste microcontrolador solo entra en funcionamiento cuando el sistema se conecta vía USB y por ende es alimentado a través de él.

La fuente tiene como elemento principal el LM1085-IT05 un regulador de voltaje de 5 Voltios cuyas características preponderantes son: soportar un voltaje de entrada de hasta 25 V, proveer una salida de corriente de hasta 3A.

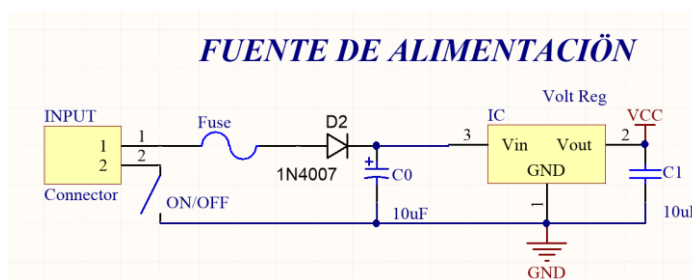


Ilustración 20- Fuente de Alimentación

El regulador se utilizó como convertidor DC-DC para proveer energía al módulo GPS, al microcontrolador Atmega328p y la pantalla LCD. El sistema cuenta con un jumper que permite seleccionar el tipo de alimentación que proveerá al circuito, ya sea a través del conector de 12 V o por medio del puerto USB.

5.2.1.8 Diseño de la Placa de Circuito Impreso PCB

Una vez se tiene el diagrama esquemático completo se procede a transferir el diseño al editor de PCB, este proceso se realiza en el editor del diagrama esquemático mediante la opción Design » Import Changes from.. .PcbDoc. Una vez seleccionada la opción los componentes y redes aparecerán en el espacio de trabajo del PCB para ser posicionadas y ruteadas.

El editor de PCB posee una opción de auto posicionamiento y auto ruteo, sin embargo se ha optado por la posición y ruteo manual para agrupar la mayor cantidad de componentes en el menor lugar posible y ubicar a los botones y los leds indicadores en una posición estratégica dejándolos expuestos para la manipulación de los usuarios finales.

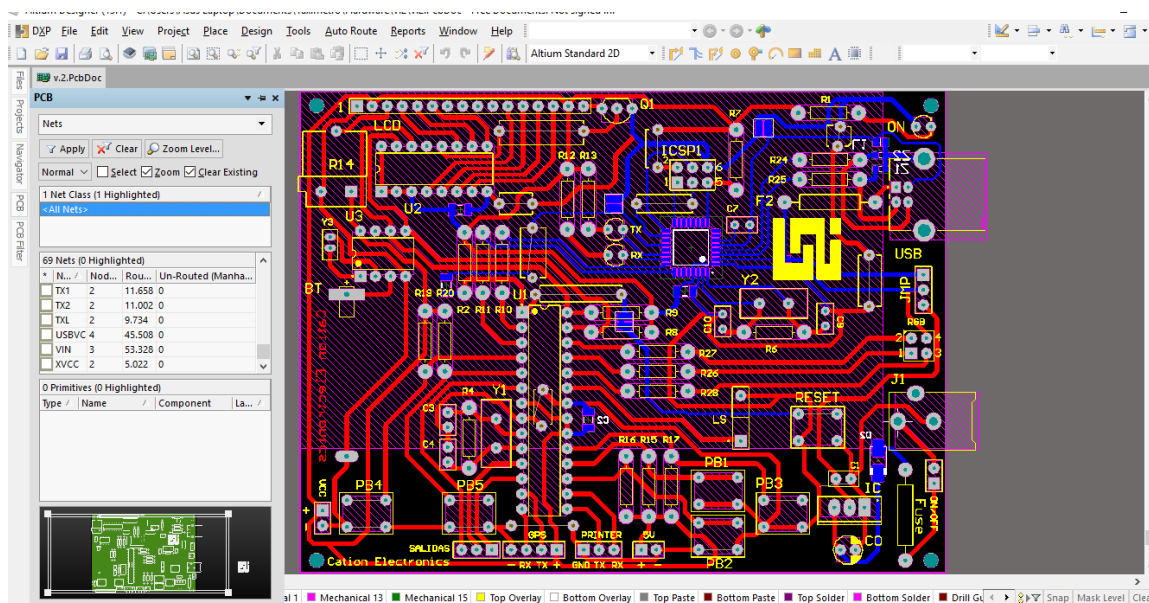


Ilustración 21- PCB del Taxímetro en el Editor PCB de Altium Designer.

Sumado a la opción de auto posicionamiento y auto ruteo el editor de PCB ofrece una gran cantidad de reglas que pueden modificarse para facilitar el trazado de las rutas, el ancho de las pistas, el uso de vías, etc.

En la capa superior de la placa (la que no tiene cobre) se colocan los componentes de montaje pasante y se sueldan del lado del cobre (capa inferior) donde se encuentran los PADS, por ello las líneas asociadas a estos componentes deben ser trazadas en la capa inferior de la placa. Distinto es el caso de los componentes de montaje superficial (SMD) ya que estos deben colocarse y soldarse del lado del cobre, las líneas asociadas a estos componentes deben ser trazadas en la capa superior de la placa.

Lo antes mencionado permite colocar ambos encapsulados en una PCB de simple cara tomando en cuenta que el diseño contempla componentes con encapsulado SMD para reducir el tamaño de la placa.

El diseño final del taxímetro contempla la necesidad de actualizar la tarifa a través de un puerto USB, pero es importante tomar en cuenta que la actualización se deber realizar por un ente regulador, por lo cual es innecesario que los taxímetros integren el controlador USB; por ello se decidió realizar 2 PCBs, una tarjeta principal, que se integra al taxímetro; y una tarjeta de comunicación USB, que es manipulada por el ente regulador para actualizar la tarifa.

Esta configuración permite ahorrar costos al momento de fabricar los taxímetros, ya que por cada 100 tarjetas principales fabricadas se puede fabricar 5 tarjetas de comunicación USB.

La tarjeta principal tiene unas dimensiones de 100mm x 60mm. A continuación se muestran las vistas en 3d de la tarjeta principal.

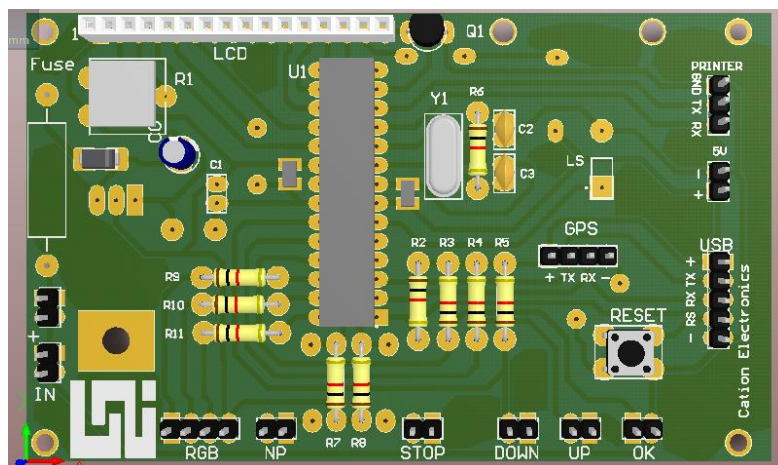


Ilustración 22- Vistas Frontal en 3D de la PCB.

La tarjeta de comunicación USB tiene unas dimensiones de 85mm x 30mm. A continuación se muestran las vistas en 3d.

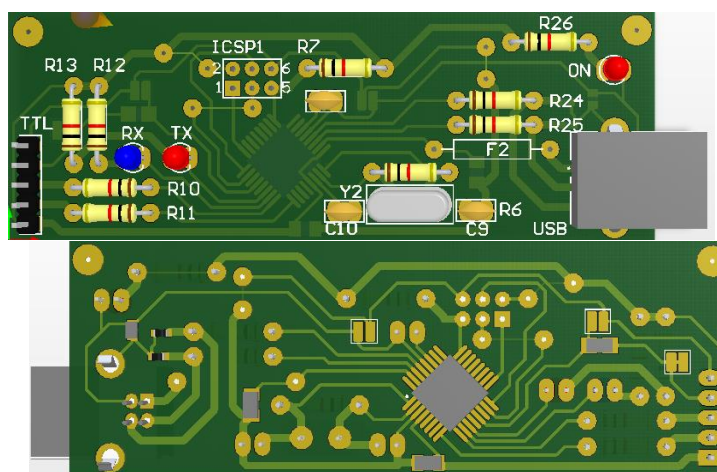


Ilustración 23- Vistas Frontal y Trasera en 3D de la tarjeta USB.

Ambas tarjetas se comunican por medio de un cable de 5 vías.

5.2.2 Desarrollo de Firmware

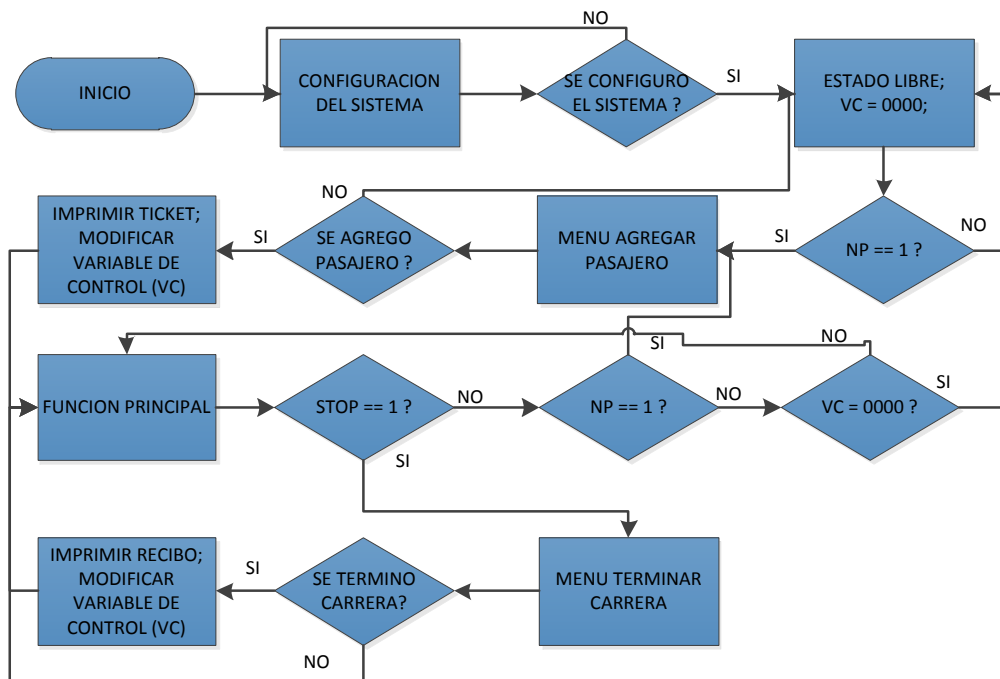


Ilustración 24- Diagrama de Flujos del Firmware

El firmware se ha escrito de forma tal que el sistema funcione de la siguiente manera: al energizar el circuito el sistema se auto configura monitoreando la señal del GPS, evalúa la calidad de la señal y decodifica los parámetros necesarios (Latitud, Longitud, Hora y fecha, Velocidad). En ausencia de señal proveniente del GPS, el sistema se bloquea hasta obtener una buena recepción de la información.

Una vez decodificados los datos del GPS el sistema procede a la función Estado Libre. Si un pasajero es agregado presionando el botón NP (Nuevo Pasajero), el sistema despliega un menú en el cual se puede navegar a través de los botones direccionales UP (direccional hacia arriba) y DOWN (direccional hacia abajo). Luego se puede seleccionar la opción deseada con el botón OK (ingresar a la opción).

Agregado el pasajero se actualiza la variable de control (VC) y se procede a imprimir un ticket, el cual le indica cual pasajero es al abordar la unidad para diferenciarse de los demás pasajeros que puedan abordar durante la trayectoria. De esta misma forma se pueden ir agregando pasajeros hasta alcanzar un máximo de 4 pasajeros simultáneos.

El sistema calcula la distancia recorrida por cada pasajero independiente y la distancia compartida (si existiera el caso) en la Función Principal. Esto permite calcular una tarifa individual por cada pasajero que aborde la unidad. Cuando haya distancia compartida el precio se fijará mediante un promedio entre el número de pasajeros que recorrieron esta distancia compartida.

Para detener la carrera es necesario presionar el botón STOP (Detener carrera), y al igual que al momento de agregar pasajero, el sistema despliega un menú donde el usuario debe seleccionar qué carrera desea detener de todas las opciones que sean posibles. El cliente que desea finalizar su carrera debe mostrar el ticket que le fue entregado inicialmente para corroborar qué pasajero es. Una vez seleccionado el pasajero que termina el trayecto, se procede a imprimir el recibo con sus respectivas cuentas.

5.2.2.1 Atmel Studio 6.2

El firmware fue desarrollado con el software original de Atmel, Atmel Studio 6.2, añadiéndole un plugin llamado Visual Micro para añadir funcionalidades de la plataforma Arduino. Atmel Studio es una plataforma de desarrollo integrada de Atmel que provee un moderno y poderoso ambiente para el desarrollo de tecnologías AVR y ARM. Permite programar y depurar ya que posee herramientas como el Ensamblador y Simulador AVR, un editor, un compilador, etc.

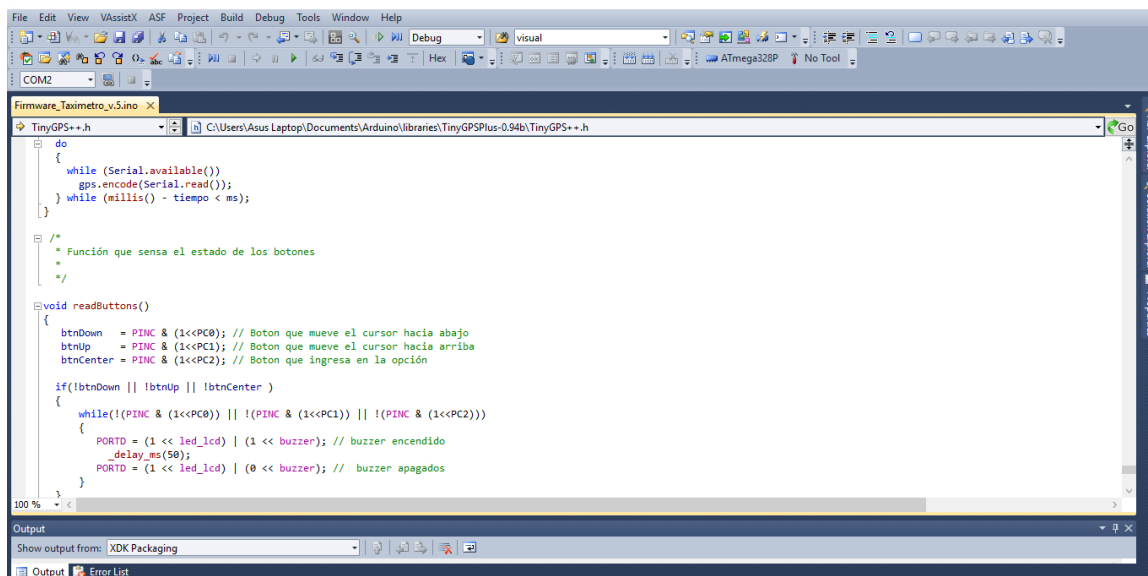


Ilustración 25- Atmel Studio 6.

En Atmel Studio los códigos presentan diferentes colores dentro de sus líneas y directivas que permiten una mejor comprensión del mismo. Aún más, esta plataforma tiene la opción de visualizar el código a manera de menú donde se puede minimizar lo que no se desee ver y maximizar aquello que se quiere observar; esto permite una rápida y fácil comprensión y navegación dentro del código.

El código del sistema se desarrolló en un lenguaje de alto nivel usando Atmel Studio, siendo este código un híbrido entre C y C++, utilizando funciones propias de cada una de éstas.

5.2.2.2 Estado Configuración del Sistema

En esta sección es necesario decodificar la señal proveniente del GPS, y configurar todos los parámetros necesarios para el correcto funcionamiento del sistema. A continuación se muestra el diagrama de flujos del algoritmo utilizado:

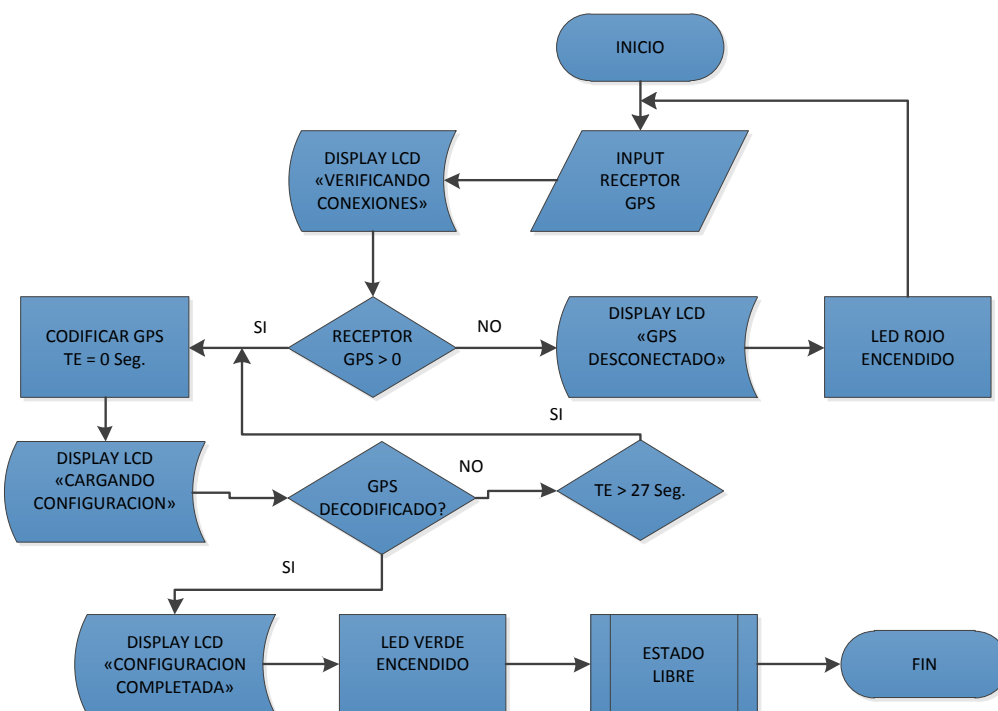


Ilustración 26- Diagrama de Flujos Estado Configuración del Sistema

Lo primero que realiza el programa es detectar la señal del GPS mediante los datos presentes en el buffer del puerto serial; si en 5 segundos no hay datos en el buffer se muestra en el LCD “GPS no detectado” y encendemos el led rojo como alarma. Si esto ocurre es probable que el receptor del GPS se encuentre desconectado o dañado y es necesaria su revisión para su respectivo diagnóstico.

Una vez detectada la presencia de datos en el buffer del puerto serial, se procede a decodificar las tramas provenientes del GPS; si han pasado 27 segundos (COLD START) y aún no se han recibido todos los datos necesarios para la decodificación, el sistema se reinicia y espera otros 27 segundos, y así sucesivamente hasta que se reciba toda la información necesaria. El tiempo de adquisición de datos varía según el grado de interferencia en la línea de vista del receptor hacia los satélites.

Es importante aclarar que éste proceso solo se realiza la primera vez que se enciende el sistema, ya que el receptor GPS cuenta con una memoria EEPROM y una batería que guardan la última posición, función que les permita detectar con mayor rapidez la siguiente señal.

Cuando la señal del GPS es decodificada, se procede a mostrar en Pantalla “Configuración Completada”, al igual que encendiendo el led verde y un doble pitido del buzzer, para luego ser remitido al estado LIBRE.

5.2.2.3 Estado Libre

El Estado Libre indica que el sistema cuenta con todos los parámetros necesarios para admitir una nueva carrera. A continuación se muestra el algoritmo utilizado:

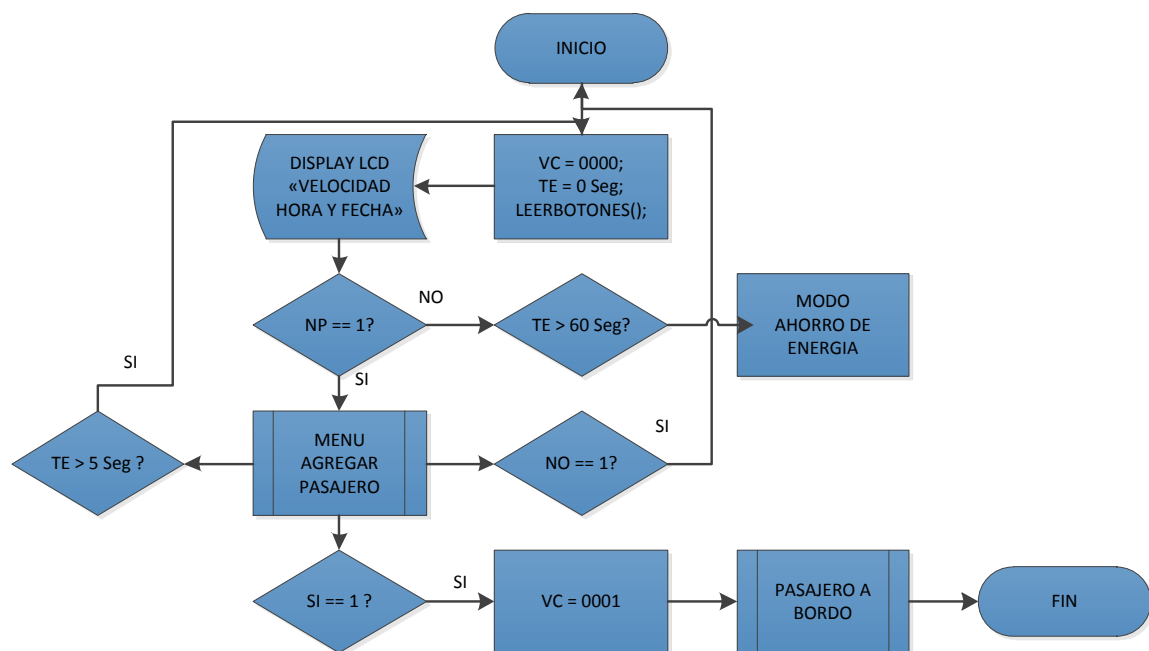


Ilustración 27- Diagrama de Flujos Estado Libre

Para manejar el estado de los pasajeros se utilizó una variable de control declarada como un nibble (siendo el bit menos significativo la representación del

pasajero 1 y el bit más significativo la representación del pasajero 4), dónde un 0 indica libre y un 1 indica ocupado. Un 1111 representa a los 4 pasajeros ocupados, mientras un 0000 todos los pasajeros libres.

Inicialmente, en ausencia de pasajeros y con nuestra variable de control (VC) en 0000, el sistema muestra en pantalla datos de interés como: Hora, fecha y velocidad del vehículo. Después de 1 minuto el sistema entra en modo de ahorro de energía, poniendo el microcontrolador en stand by y apaga el backlight del LCD. Para salir del modo de ahorro de energía es necesario presionar uno de los botones.

Si el botón nuevo pasajero (NP) es presionado, se procede a desplegar el menú agregar pasajero.

5.2.2.4 Menú Agregar Pasajero

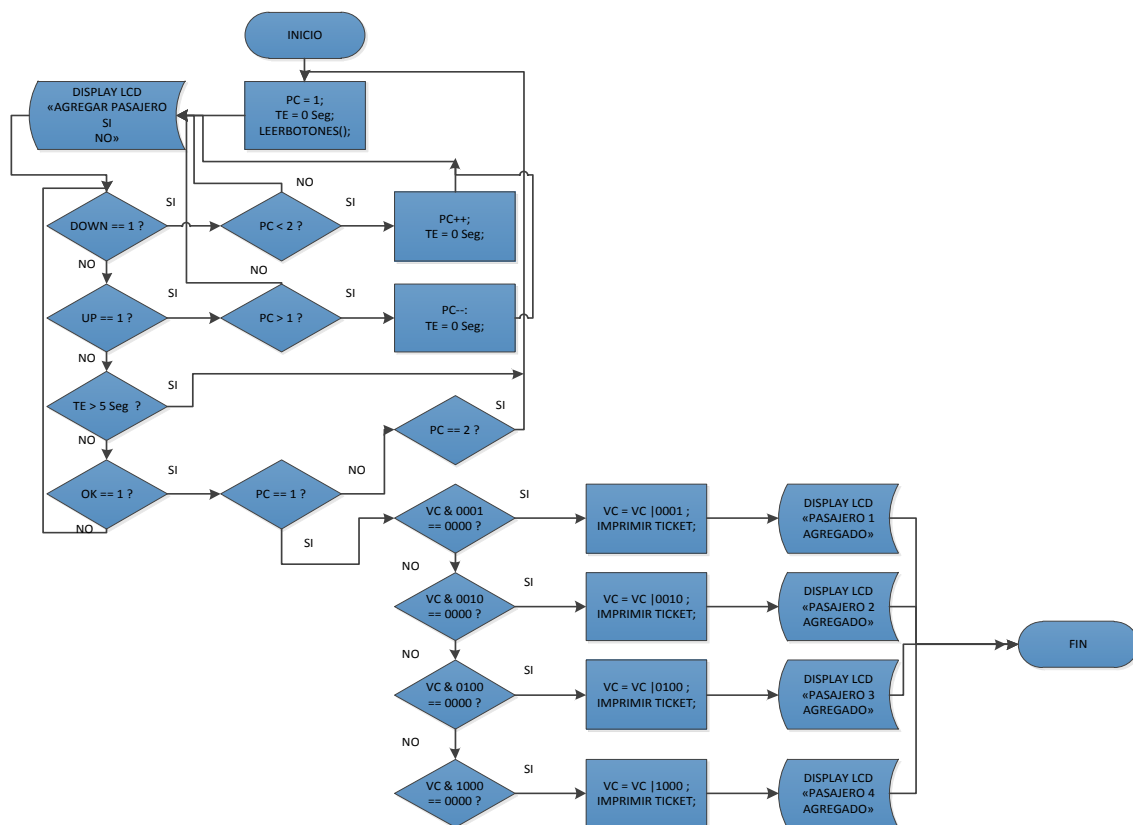


Ilustración 28- Diagrama de Flujos Menú Agregar Pasajero

Si el botón nuevo pasajero es presionado, el sistema despliega un menú dónde el usuario confirma o rechaza la petición.

Dentro de este menú el usuario puede deslizarse a través de un cursor (PC) controlado por las direccionales UP y DOWN y seleccionar la opción deseada con el botón INTRO/OK. Si el usuario rechaza la petición de agregar pasajero es remitido al Estado Libre o si ya hay pasajero a bordo es devuelto a la posición anterior. Si el usuario confirma la petición de agregar pasajero, la variable de control toma un nuevo valor que es 0001 (pasajero 1 agregado).

Si el botón de nuevo pasajero es presionado nuevamente y la solicitud es aceptada, la variable de control cambia su valor de 0001 a 0011, indicando un nuevo estado con un segundo pasajero a bordo. El valor de la variable de control para el pasajero 3 es 0111 y para el pasajero 4 es 1111. Cada vez que un pasajero es agregado se procede a imprimir un ticket que identifica a cada pasajero si se cuenta con la impresora; si no se cuenta con la impresora se procede a imprimir en pantalla que el pasajero ha abordado la unidad, con el número respectivo y con el color de ficha que se le ha de entregar para poder identificarlo.

➤ Fichas Identificativas

En caso de no tener impresora es necesario tener otra manera de identificar al pasajero y se decidió hacerlo a través de fichas de colores, cuatro distintas, una para cada uno. Los colores son:

Color Ficha	Pasajero
Rojo	Pasajero 1
Amarillo	Pasajero 2
Verde	Pasajero 3
Azul	Pasajero 4

Tabla 8- Asignación de Fichas Identificativas

Para poder llevar a cabo los cambios de estado en la variable de control desde 0000 hasta 1111 se necesita una máscara y consiste en la comparación del nibble con otro nibble predeterminado utilizando funciones como AND y OR para validar el estado de la variable. El programa realiza de manera cíclica y automática la asignación del número del pasajero que está abordando gracias a esta máscara. El proceso es a como sigue:

Inicialmente la variable de control es 0000. Al agregar pasajero el programa compara este valor usando la función AND con el valor 0001. El resultado a esto es el valor 0000 que indica que no hay nadie dentro de la unidad. Para modificar este valor luego se realiza una comparación usando la función OR con el valor 0001 nuevamente. El resultado es 0001 y la variable de control adopta este nuevo valor indicando su nuevo estado: pasajero 1 a bordo.

Si un nuevo pasajero desea ser agregado la variable de control, ahora 0001, es comparada con 0001 usando la función AND. Su resultado es 0001, por tanto hay 1 pasajero ya en la unidad. Ahora se aplica otra AND con 0010 para verificar si hay algún pasajero 2. El resultado es 0000. Esto indica que los 3 pasajeros restantes están libres, por lo cual se puede agregar un nuevo pasajero. Para agregarlo se procede a comparar el valor mediante una OR con 0010 cuyo resultado es 0011. Ahora hay 2 pasajeros a bordo.

Nuevamente el botón de nuevo pasajero es presionado y se acepta la petición. El programa ahora compara el nuevo valor 0011 con 0001 mediante una AND. Su resultado es 0001, o sea, hay pasajero 1 en la unidad. Luego comparamos 0011 con 0010 mediante una AND. Su resultado es 0010, es decir, hay pasajero 2 en la unidad. De nuevo se compara 0011 mediante una AND pero ahora con 0100 para verificar si hay pasajero 3. El resultado es 0000, por tanto, no hay pasajero 3. Se procede a agregar el pasajero usando una OR entre el valor 0011 y 0100, cuyo resultado es 0111 indicando un tercer pasajero a bordo.

En este punto se pretende agregar un nuevo pasajero. Ahora la variable de control es 0111. Se aplican ANDs de manera consecutiva para verificar el número de pasajeros actualmente en la unidad.

$0111 \& 0001 = 0001$ (hay pasajero 1).

$0111 \& 0010 = 0010$ (hay pasajero 2).

$0111 \& 0100 = 0100$ (hay pasajero 3).

$0111 \& 1000 = 0000$ (no hay pasajero 4).

Luego aplicamos una OR para modificar el valor de la variable entre 0111 y 1000. Nuestro nuevo valor ahora es 1111: cuatro pasajeros a bordo.

Si se desea agregar un quinto pasajero el programa verifica el valor de la variable de control. Si su valor es igual a 1111, como es el caso, no agrega pasajeros ni aplica ningún tipo de función sino que manda a imprimir en pantalla “No puede agregar más pasajeros”.

5.2.2.5 Función Principal (Pasajero a Bordo)

Una vez dentro de la función Pasajero a Bordo el sistema inicia a medir la distancia recorrida por el pasajero a través del GPS de manera continua. La distancia es representada por la variable d y es enumerada según el pasajero (d1 para pasajero 1, d2 para pasajero 2, y así sucesivamente). Además, hay una variable de referencia llamada “odómetro” que incrementa su valor por cada segundo; esta variable es la que permite la función de multipasajeros. Este es el diagrama de flujo correspondiente:

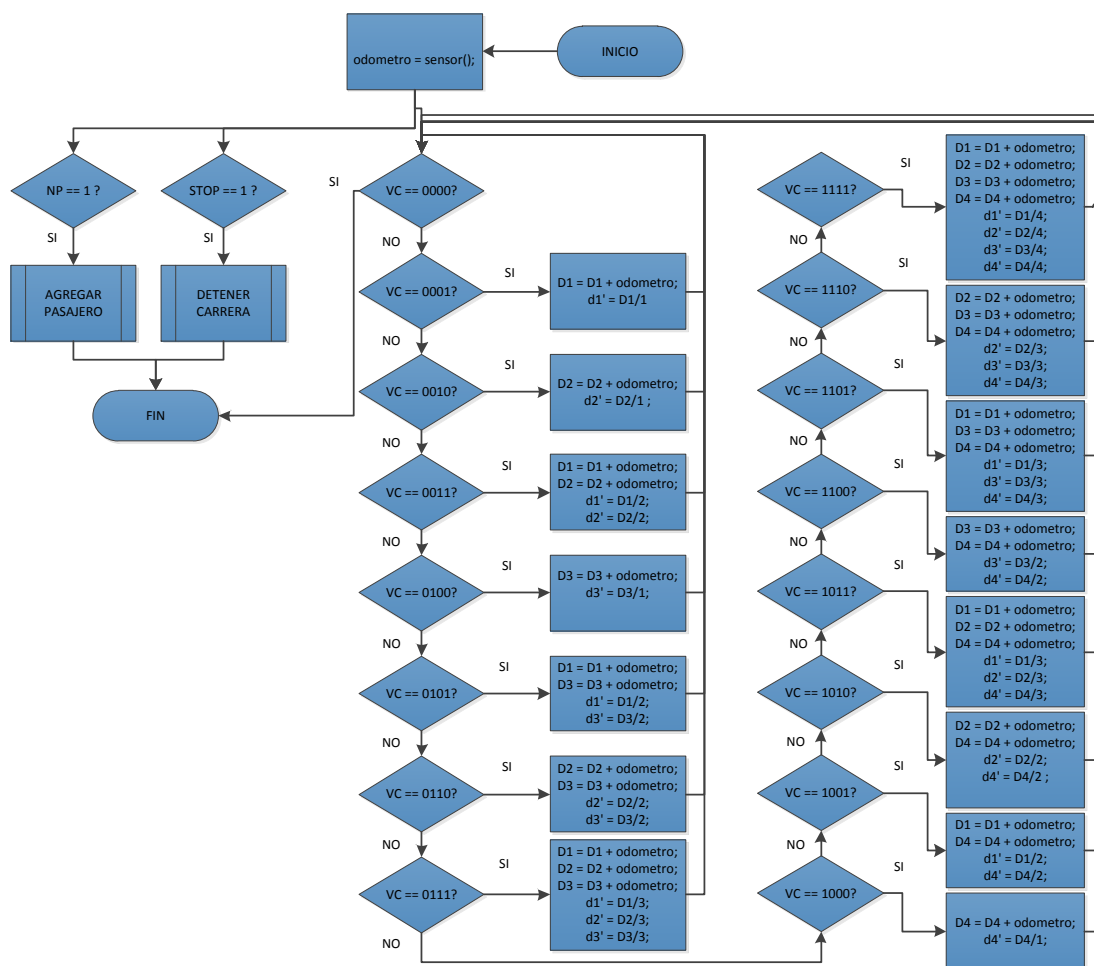


Ilustración 29- Diagrama de Flujos Función Principal

La variable “odómetro” inicialmente se encuentra en 0 Km, así como d1, d2, d3 y d4. Una vez que se agrega el pasajero 1 (VC = 0001), “odómetro” aumenta gradualmente su valor cada segundo en función del cuentakilómetros del GPS (sensor ()), al igual que d1. El valor d1 será igual a “odómetro”. Si se recorren 2 Km entonces “odómetro” vale 2 Km y d1 también.

Ahora se procede a agregar el pasajero 2 (VC = 0011), cuyo valor respectivo d2 es 0 Km inicialmente. La variable “odómetro” se coloca a 0 Km nuevamente y d1 mantiene su valor anterior de 2 Km. El valor d2 incrementará en igual medida que “odómetro” al igual que d1 (recuérdese que d1 = 2 Km). Si se recorren otros

2 Km entonces “odómetro” vale 2 Km, d2 vale 2 Km, y d1 vale 4 Km (2 Km anteriores más 2 Km actuales).

Luego un pasajero 3 decide abordar la unidad (VC = 0111). La variable “odómetro” nuevamente se inicializa a 0 Km, d3 vale 0 Km, d2 vale 2 Km y d1 vale 4 Km. Ahora d3 aumentará su valor en la misma proporción que “odómetro”, al igual que d2 y d1 (recuérdese que d2 = 2 Km y d1 = 4 Km). Otros 2 Km se recorren, por tanto “odómetro” es 2 Km, al igual que d3. Sin embargo d2 vale ahora 4 Km (2 Km anteriores más 2 Km actuales) y d1 es igual a 6 Km (4 Km anteriores más 2 Km actuales).

Después un pasajero 4 desea abordar la misma unidad (VC = 1111). Nuevamente igualamos “odómetro” a 0 Km y d4 inicializa en 0. Sin embargo d3 es igual a 2 Km, d2 igual a 4 Km y d1 igual a 6 Km. Tanto d4 como d3, d2 y d1 aumentarán al mismo ritmo que “odómetro”. 2 Km adicionales se recorren con los cuatro pasajeros a bordo, por lo cual “odómetro” toma el valor de 2 Km, así como d4. Pero ahora d3 es igual a 4 Km (2 Km anteriores más 2 Km actuales), d2 es igual a 6 Km (4 Km anteriores más 2 Km actuales) y d1 es igual a 8 Km (6 Km anteriores más 2 Km actuales).

Supóngase que el pasajero 3 finaliza su recorrido en este punto (VC = 1011). Entonces se le brinda su tarifa a pagar en base a sus 4 Km (virtualmente recorrió menos y pagará menos que 4 Km, más adelante veremos por qué) y luego d3 cambia a su valor inicial de 0 Km pero d4, d2 y d1 mantienen sus valores de 2 Km, 6 Km y 8 Km respectivamente, valores que igualmente seguirán aumentando en igual medida que “odómetro”.

De esta manera es que la variable de referencia “odómetro” es utilizada para hacer que el sistema cumpla con la función de multipasajeros, donde cada carrera es independiente de las demás.

Ahora bien, el sistema es capaz de medir las distancias de manera independiente para que el usuario sepa cuánto ha recorrido en total pero además es capaz de conocer las distancias recorridas sin compartir la unidad y compartiendo la unidad. Como se ha explicado anteriormente en la sección “Modelo Tarifario” el sistema debe hacer un reajuste en la tarifa final cuando el vehículo transporte a más de un usuario a la vez. Una carrera de 2 Km de un solo usuario no es equivalente a la misma carrera realizada por 2 o más usuarios a la vez. En el primer caso el usuario único debe cubrir con el gasto total de la carrera pero en el segundo caso el gasto total debe ser distribuido entre el número de usuarios.

Como se mencionó anteriormente en la sección de Modelo Tarifario: “Así, si un usuario recorre 5 Km de una carrera de 10 Km solo, debe cubrir con el costo equivalente a esos 5 Km. Pero si un segundo usuario aborda la unidad en ese momento y se dirige al mismo punto que el primer usuario, entonces el costo de los siguientes 5 Km será compartido entre ambos usuarios. Al final el primer usuario pagará 5 Km independientes más 5 Km compartidos entre 2 usuarios y el segundo usuario únicamente pagará los 5 Km compartidos entre 2 usuarios”.

Para realizar esto el sistema utiliza otras variables para cada carrera, que son: $d1'$, $d2'$, $d3'$, $d4'$. Estas variables representan las distancias “virtuales” que ha recorrido cada pasajero. Les llamamos virtuales porque al compartir la unidad de transporte el valor real de las distancias se disminuye en base a lo explicado en el “Modelo Tarifario”.

Para medir esta distancia virtual el sistema se auxilia de la variable de control VC que muestra el número de pasajeros a bordo y de “odómetro”. Al igual que las distancias reales, las distancias virtuales incrementan conforme a “odómetro” pero con la diferencia de que ahora será dividida entre el número de pasajeros y su resultado se sumará a su valor anterior y VC es la que indica el número de

pasajeros (Ejemplo: 1111, 4 pasajeros a bordo; 0000, 0 pasajeros a bordo; 1010, 2 pasajeros a bordo).

Tomemos este ejemplo: La variable “odómetro” inicialmente se encuentra en 0 Km, así como $d1'$, $d2'$, $d3'$ y $d4'$. Una vez que se agrega el pasajero 1, “odómetro” aumenta gradualmente su valor cada segundo en valores de “odómetro” / 1 (pasajero), al igual que $d1'$. El valor $d1'$ será igual a “odómetro”. Si se recorren 2 Km entonces “odómetro” vale 2 Km y $d1'$ también, porque “odómetro” está siendo dividida únicamente entre 1 pasajero.

Ahora se agrega el pasajero 2, $d2'$ es 0 Km. La variable “odómetro” se coloca a 0 Km; $d1'$ mantiene su valor anterior de 2 Km. El valor $d2'$ incrementará por “odómetro” / 2 (pasajeros), al igual que $d1'$ a partir de este momento (recuérdese que $d1' = 2$ Km). Si se recorren otros 2 Km entonces el aumento es $2 \text{ Km} / 2 = 1$ Km. Por tanto $d2'$ aumentó su valor en 1 Km (distancia virtual) en vez de 2 Km (distancia real), y $d1'$ vale ahora 3 Km (2 Km anteriores más 1 Km actual).

Luego un pasajero 3 aborda la unidad. Variable “odómetro” nuevamente a 0 Km, $d3'$ vale 0 Km, $d2'$ vale 1 Km y $d1'$ vale 3 Km. Ahora $d3'$ aumentará su valor por “odómetro” / 3 (pasajeros), al igual que $d2'$ y $d1'$ a partir de este momento (recuérdese que $d2' = 1$ Km y $d1' = 3$ Km). 3 Km se recorren, por tanto el incremento es $3 \text{ Km} / 3 = 1$ Km. Entonces ahora $d3'$ vale 1 Km (distancia virtual) y no 3 Km (distancia real). Sin embargo $d2'$ vale ahora 2 Km (1 Km anterior mas 1 Km actual) y $d1'$ es igual a 4 Km (3 Km anteriores mas 1 Km actual).

Después un pasajero 4 hace uso de la misma unidad. Entonces “odómetro” regresa a 0 Km y $d4'$ vale 0 Km también. El aumento ahora será igual a “odómetro” / 4 (pasajeros). Sin embargo en este punto $d3'$ es igual a 1 Km, $d2'$ igual a 2 Km y $d1'$ igual a 4 Km. Supónganse ahora 4 Km más de recorrido. El incremento es igual a $4 \text{ Km} / 4 = 1$ Km. Quiere decir que $d4'$ toma el valor de 1 Km (distancia virtual) en vez de los 4 Km (distancia real), $d3'$ vale 2 Km (1 Km

anterior más 1 Km actual), d2' vale 3 Km (2 Km anteriores más 1 Km actual) y d1' vale 5 Km (4 Km anteriores más 1 Km actual).

Finalmente el pasajero 1 termina su carrera y desea conocer su tarifa. Realmente recorrió 2 Km (solo) + 2 Km (pasajero 1 con pasajero 2) + 3 Km (pasajero 1 con pasajero 2 y 3) + 4 Km (pasajero 1 con pasajero 2, 3 y 4) pero eso no indica que el sistema le facturará 11 Km, sino 2 Km (solo) + 1 Km (2 Km / 2 pasajeros) + 1 Km (3 Km / 3 pasajeros) + 1 Km (4 Km / 4 pasajeros), que sería un total de 5 Km.

De esta manera el sistema brinda una tarifa basada en la distancia compartida por los usuarios para hacerlo justo tanto para el cadete como para el usuario. Apoyado en las variables d1', d2', d3', d4' y VC para conocer el número de pasajeros a bordo se cumple con lo establecido en el Modelo Tarifario que se ha propuesto.

Cabe destacar que el programa principal es el que lleva el control de las distancias reales y virtuales, pero quien se encarga de brindar la tarifa es el Menú Finalizar Carrera. Una vez que algún usuario decide finalizar carrera se hace una interrupción en el programa principal y se ingresa a dicho menú, explicado en las siguientes páginas.

5.2.2.6 Menú Terminar Carrera

Una vez dentro del menú el usuario puede desplazarse mediante un cursor (PC) a través de las direccionales UP y DOWN y seleccionar la opción deseada con el botón INTRO/OK. A diferencia del Menú Agregar Pasajero donde el sistema asigna de manera automática el número de pasajero, en el Menú Terminar Carrera es el conductor del vehículo el que selecciona al pasajero en base al ticket impreso o en base a la ficha recibida por el pasajero al iniciar su carrera. Para llevar a cabo esto se debe auxiliarse de la misma variable de control VC, el

nibble que indica el número de pasajeros a bordo. Esta es la lógica que sigue el menú:

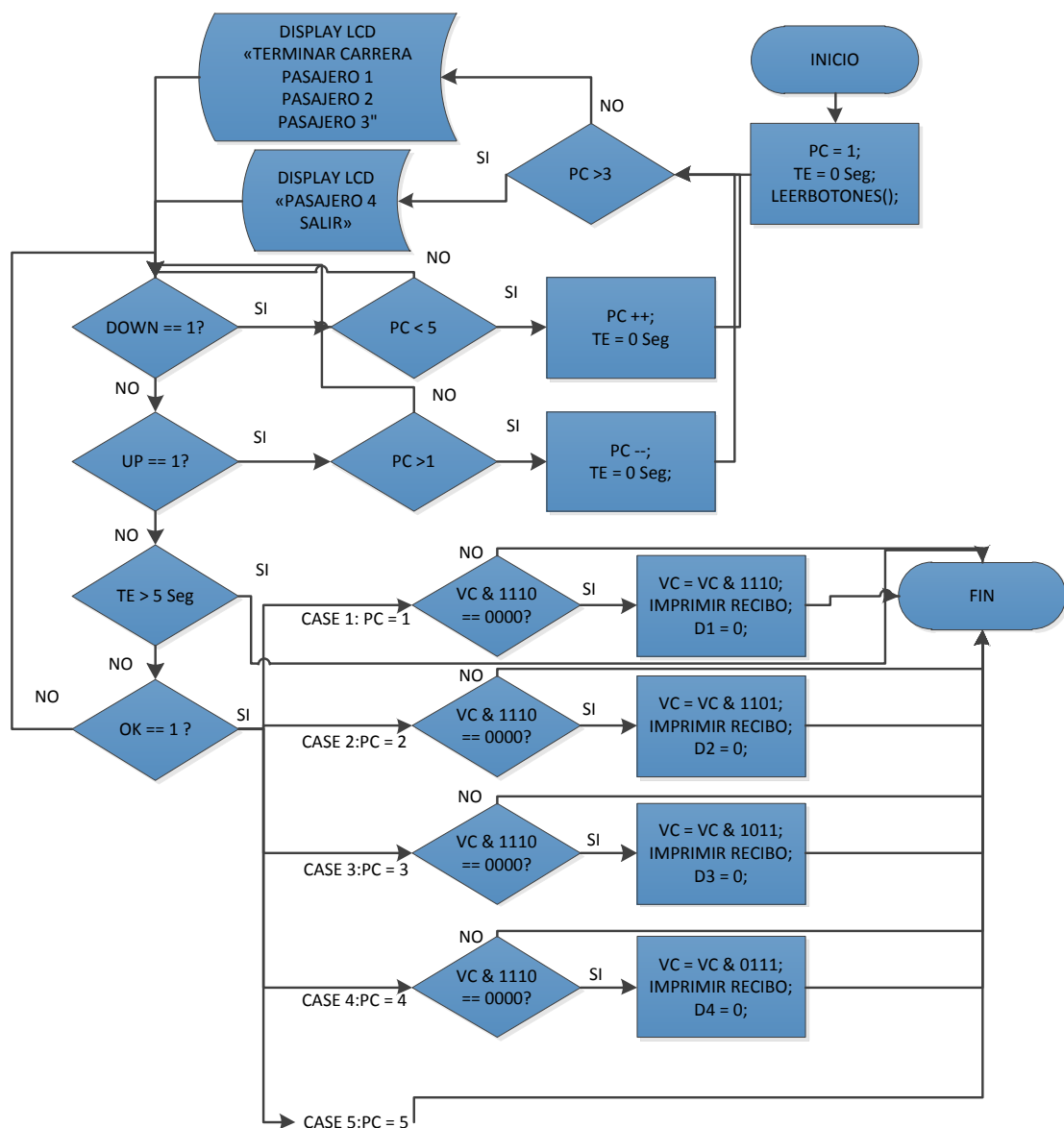


Ilustración 30- Diagrama de Flujos Menú Terminar Carrera

Primeramente el sistema verifica el estado mediante el nibble: si el usuario apunta a una posición donde no hay pasajero entonces el sistema regresa a seguir ejecutando su función principal; si el usuario apunta a una posición donde sí hay pasajero entonces finaliza esa carrera específica. Para esto se utiliza la función AND.

Considérese el estado 1101 donde el usuario desea finalizar la carrera del pasajero 2. Esta carrera no se está llevando a cabo ya que su valor correspondiente dentro del nibble es igual a 0 (1101). El sistema verifica el estado aplicando una AND al nibble, utilizando el valor 0010. El resultado es 0 para el bit deseado, lo que indica que no se está ejecutando ninguna carrera en esa posición.

Ahora considérese el mismo estado 1101 donde el usuario desea finalizar la carrera del pasajero 1. Esta carrera si se está realizando, el valor 1 del bit correspondiente lo indica (1101). Primero se verifica el estado con la función AND, usando ahora el valor 0001 (por lo que es el bit menos significativo el que estamos evaluando). El resultado para el bit deseado es 1, lo que significa que si se está llevando a cabo esta carrera.

Una vez verificado el estado se detiene la carrera, se procede a facturar y luego se cambia el valor de VC aplicando nuevamente una AND pero de manera inversa (Función NOT) sobre el bit deseado (1101). Nuestro estado evaluado fue 1101 y se pretende cambiar el bit correspondiente al pasajero 1 (1101), por tanto se aplica la función AND entre 1101 y 1110, cuyo resultado para el bit deseado es 0 (1100). De esta manera se modifica VC de 1101 a 1100, se devuelve este valor de VC a la función principal y con esto se indica que el pasajero 1 ha finalizado su carrera.

En cualquiera de los demás casos posibles el procedimiento es el mismo. Si se desea evaluar y cambiar el estado correspondiente al pasajero 4 se considera el bit más significativo (1111). Si fuese el pasajero 3 se considera el siguiente bit hacia la derecha (1111). Para el pasajero 2 se considera el siguiente bit hacia la derecha luego del último mencionado (1111). Y finalmente para el pasajero 1 se considera el bit consecuente, el bit menos significativo (1111).

Para facturar las carreras primero se verifica si hay algún pasajero en la posición indicada y de ser así, se recibe el valor de la distancia virtual correspondiente y se aplica la fórmula propuesta en el Modelo Tarifario.

Una vez que se conoce el valor a pagar se ejecuta la instrucción de imprimir recibo si se cuenta con la impresora o de imprimir en pantalla en caso de no tener la impresora.

5.2.2.7 Medición de Distancia

Para calcular la distancia recorrida por el vehículo se creó una función que, incrementa el valor de la distancia en función del cambio de posición (Latitud y Longitud) que proporciona el GPS cada segundo. La función calcula la distancia entre los puntos de posición que proporciona el GPS cada segundo utilizando trigonometría esférica considerando a la Tierra totalmente esférica con la fórmula de Harvesine y luego realiza una sumatoria de todos los valores parciales, esta sucesión de puntos nos describe la trayectoria seguida por el vehículo.

$$\text{Ec. (5) } d = \sum 2 * r * \arcsin \left(\sqrt{\sin^2 \frac{\text{lat2}-\text{lat1}}{2} + \cos(\text{lat1}) * \cos(\text{lat2}) \sin^2 \frac{\text{long2}-\text{long1}}{2}} \right)$$

Donde:

d = distancia en metros entre dos coordenadas según la fórmula de Harvesine.

r = radio de la tierra (6372795 metros).

lat1 = latitud 1

lat2 = latitud 2

long1 = longitud 1

long2 = longitud 2

Es importante mencionar que la tierra no es totalmente esférica, por ello se puede obtener un error de 0.5 % en el cálculo total.

5.2.2.8 Cálculo de la Tarifa.

Para el cálculo de la tarifa el sistema toma el valor de distancia recorrida y lo multiplica por la tarifa variable por distancia y al resultado se le suma la tarifa inicial. El sistema redondea el precio de la tarifa a los 5 córdobas más próximos.

5.2.2.9 Programación De Los Microcontroladores

Hay una variedad de herramientas de programación por hardware para los microcontroladores AVR, pero si el chip tiene un gestor de arranque (Bootloader) específico estos pueden ser programados por medio de los puertos seriales sin la necesidad de tener un hardware externo.

El Bootloader es un pequeño software, que reprograma el MCU para guardar en la memoria flash el código hexadecimal a través del puerto serial, con la desventaja que el lugar que ocupa en memoria (2KB en el Atmega328P) no puede ser usado por el programa principal.

5.2.2.9.1 Conector Para El Programador ICSP

Para cargar el firmware en los microcontroladores se utilizó el puerto ICSP, que cuenta con los terminales conectados al puerto SPI del microcontrolador, como lo son: RESET, SCK, MIMO, MISO, VCC y GND. Con esta configuración se puede cargar el archivo.hex a los microcontroladores. En el circuito electrónico se proporciona 1 conector ICSP para el microcontrolador ATmega8u2 que por ser SMD es imposible retirarlo del circuito para cargar el firmware en un programador externo.



Ilustración 31- Pines Conector ICSP

Para cargar el firmware en el microcontrolador ATmega328p es necesario retirarlo del circuito y montarlo ya sea en una placa de entrenamiento o en una tabla de nodos para tener acceso al puerto ISP.

5.2.2.9.2 Programación Del Microcontrolador Atmega8u2

Como se mencionó anteriormente el ATmega 8U2 funciona como un puente entre la computadora y el puerto TTL. Por tanto, es necesario cargarlo con un firmware que use los drivers estándar de los puertos USB COM.

Para cargar el firmware en el MCU se utilizó un programador llamado USBTinyISP que utiliza el puerto ISP para comunicarse con el MCU. Este programador solo se utiliza una vez para cargar el Bootloader, y luego podemos grabar los firmware directamente desde el PC por el puerto USB emulando un puerto serie.



Ilustración 32- Programador USBTinyISP.

5.2.3 Interfaz en LabView

Para la actualización de tarifa se desarrolló una interfaz visual donde el ente regulador puede por medio de la tarjeta controladora USB cambiar los parámetros como: Tarifa Inicial y Tarifa Variable.



Ilustración 33- Interfaz Visual en LabView.

Inicialmente es necesario seleccionar el puerto comm a través de una lista desplegable en la parte CONFIGURACION. Si la tarjeta es conectada correctamente la barra DISPONIBLE cambia a verde. Al contrario, si la tarjeta no es reconocida la barra DISPONIBLE mantiene un color rojo.

Una vez detectado el puerto es necesario presionar el botón Iniciar para habilitar al taxímetro en modo ACTUALIZACION. Con el taxímetro en modo ACTUALIZACION se procede a escribir las respectivas tarifas y son enviadas a través del botón actualizar. Con las tarifas actualizadas se procede a presionar el botón listo para reiniciar el taxímetro con las nuevas tarifas.

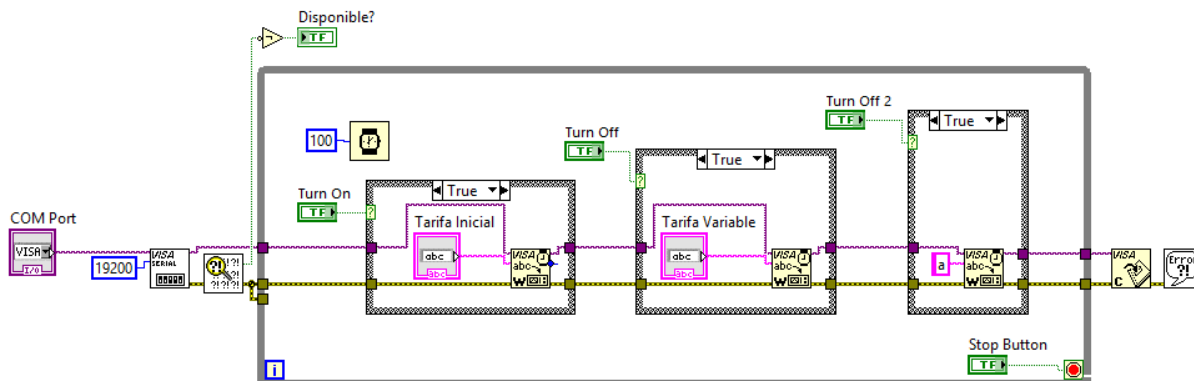


Ilustración 34- Instrumento Virtual de la Interfaz (VI).

5.2.4 Diseño de Empaquetado

El taxímetro debe ser instalado de manera estratégica dentro del vehículo que permita una clara visualización por parte de los usuarios, fácil manipulación, y sobre todo que ocupe el menor lugar posible. Es por ello que se decidió realizar esbozos utilizando herramientas informáticas para el modelado 3d del “case” del sistema.

Para el modelado 3d de la caja se utilizó el programa SketchUp en su versión gratuita. SketchUp es un programa de diseño gráfico y modelado en tres dimensiones (3D) basado en caras. Su principal característica es la de poder realizar diseños en 3D de forma extremadamente sencilla. El programa incluye entre sus recursos un tutorial en vídeo para ir aprendiendo paso a paso cómo se puede ir diseñando y modelando el propio ambiente. Permite conceptualizar y modelar imágenes en 3D de edificios, coches, personas y cualquier objeto o artículo que imagine el diseñador o dibujante, además el programa incluye una galería de objetos, texturas e imágenes listas para descargar.

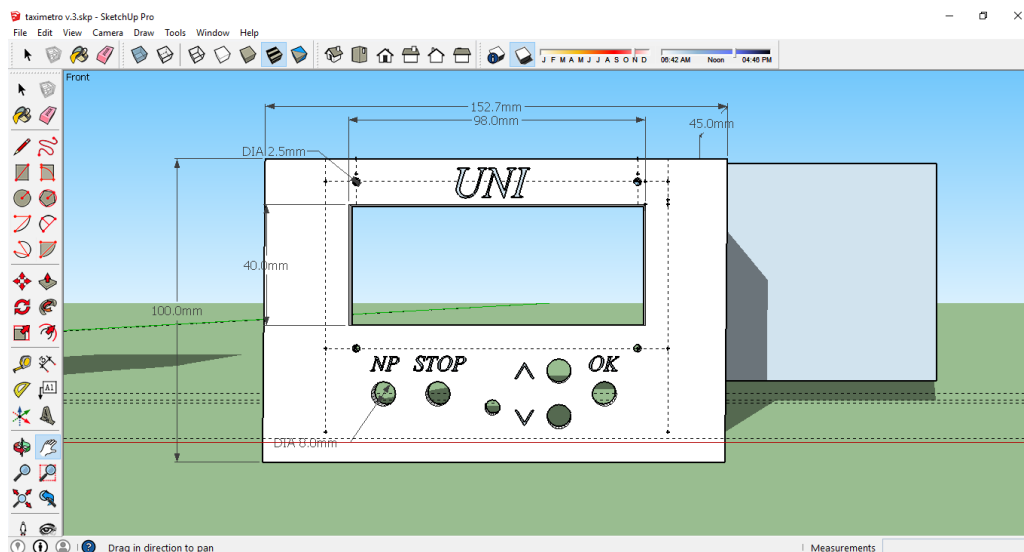


Ilustración 35- Diseño Empaquetado en SketchUp

Como se aprecia en la figura el case del dispositivo cuenta con unas dimensiones de 15cm de ancho por 10cm de alto por 6cm de profundo con una abertura rectangular de 10cm de ancho por 4cm de alto al centro superior de la parte frontal para ubicar la pantalla LCD, además de cinco orificios circulares ubicados de manera estratégica para posicionar los botones de mando. También en la parte central inferior del frente hay un pequeño orificio circular por debajo de los botones donde se ubica el LED RGB que indica el estado del sistema.

El case se diseñó de manera tal que la parte trasera pueda ser removida para poder realizar las conexiones pertinentes y para poder ensamblar y desensamblar el circuito cuando sea necesario.

6 ANALISIS DE RESULTADOS

Este apartado muestra todas las pruebas realizadas en campo del proyecto instalado en un vehículo con conector de mechero estándar de 12VDC, tomando en consideración todos los parámetros que influyen en el comportamiento del prototipo y además el análisis económico del proyecto. Este apartado se divide en 5 subsecciones que son: Prueba funcional del dispositivo, pruebas de impresión, pruebas de distancia, pruebas de costos por carrera y finalmente, análisis económico.

Para propósitos de prueba del taxímetro se utilizó un adaptador universal marca Grenosis Dual Power que consta de dos salidas, una de 2.1A conectada directamente a la impresora, y otra de 1A conectada directamente al circuito omitiendo el regulador de voltaje, ya que ambas salidas proveen 5VDC.



Ilustración 36- Adaptador Universal de Alimentación para Autos.

6.1 Prueba Funcional

Estas pruebas se basan en la parte funcional del equipo, en conocer si el diseño propuesto fue o no alcanzado con éxito, de la interacción y manejo, y de conocer si cumple paso a paso los diagramas de flujo propuestos en el apartado de diseño.

6.1.1 Recepción y Decodificación de Datos del GPS

El primer estado del prototipo es Configuración del Sistema. En esta etapa el GPS recibe las tramas de datos enviados desde los satélites hasta obtener datos válidos acerca de la posición actual. Mientras se están decodificando las tramas se muestra en la pantalla LCD “Cargando Configuración” como se muestra en la figura 38:



Ilustración 37- Pantalla Cargando Configuración

Esta prueba se realizó en carretera abierta con el taxímetro instalado en el vehículo con la antena externa del GPS ubicada en el tablero con buena línea de vista hacia los satélites, obteniendo los siguientes resultados acerca del tiempo de conexión que toma el GPS para recibir los datos válidos de posición.

PRUEBAS	RESULTADOS
Prueba 1	18 segundos
Prueba 2	15 segundos
Prueba 3	17 segundos
Prueba 4	20 segundos
Prueba 5	10 segundos
Prueba 6	15 segundos
PROMEDIO	15.8 segundos

Tabla 9- Pruebas de Tiempo de Conexión GPS en Carretera Abierta

Si existe algún problema con las conexiones del GPS al circuito se muestra en pantalla: “GPS no detectado”.

6.1.2 Verificación de Datos en Estado Libre

Una vez que el GPS ha recibido y decodificado los datos necesarios para la inicialización del sistema, se ingresa por defecto al Estado Libre.



Ilustración 38- Pantalla Estado Libre

En esta pantalla se presenta la fecha actual en formato DD/MM/AAAA. La hora actual se muestra en formato militar de 24 horas HH: MM: SS. Inicialmente la hora generada por el GPS es en estándar UTC, es decir 6 horas por delante de la hora local; por ello se procedió a realizar una corrección de horario en el firmware obteniendo los datos de zona horaria para Nicaragua (UTC-6), como son hora y fecha actual. Y finalmente en la parte inferior de la pantalla se muestra la velocidad actual del vehículo en KM/H.

6.1.3 Funcionalidad en Menú Agregar Pasajero

Para acceder al menú AGREGAR PASAJERO desde el ESTADO LIBRE se debe presionar el botón NP, dentro del menú existen 2 opciones de navegación las cuales están presididas por un cursor, como se puede observar en las fotografías de la figura 4.9.

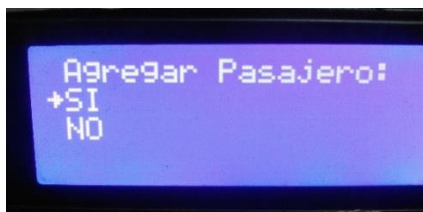


Ilustración 39- Pantalla Menú Agregar Pasajero

El cursor es controlado por los botones direccionales (arriba y abajo), y se puede acceder a las opciones de navegación a través del botón OK. Si se selecciona la opción 2 (NO) el prototipo procede al ESTADO LIBRE; sin embargo si se selecciona la opción 1 (SI), el sistema despliega otro menú en el cual el usuario puede navegar entre las 2 opciones existentes.

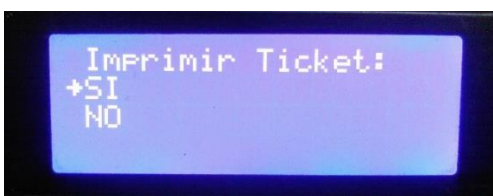


Ilustración 40- Pantalla Menú Agregar Pasajero/Imprimir Ticket

Si el usuario selecciona la opción 1 (SI) el sistema procede a imprimir el ticket y a su vez muestra en pantalla en letra grande el número de pasajero que fue agregado. Si es la opción 2 (NO) que fue seleccionada el sistema muestra en pantalla en letra grande el número de pasajero agregado y en la parte inferior izquierda el color identificativo del pasajero abordado.



Ilustración 41- Pantalla Menú Agregar Pasajero/Pasajero Agregado

El sistema se bloquea por 10 segundos y luego se procede a la función principal. Es importante recordar que dentro de la función principal también es posible acceder al menú AGREGAR PASAJERO.

6.1.4 Funcionalidad en Función Principal

En la función principal se muestra el precio en córdobas de la carrera a medida que se va avanzando sobre la marcha. El precio va actualizándose en dependencia de la distancia. Existe una línea para cada pasajero en la pantalla LCD, a como se puede apreciar en la siguiente figura:



Ilustración 42- Pantalla Función Principal

Para la prueba de la Función Principal se realizaron las 16 diferentes combinaciones en las que el estado del sistema puede estar, desde ningún pasajero a bordo hasta 4 pasajeros a bordo y las diferentes posibilidades en que estos eventos pueden ocurrir. La respuesta del sistema ante los 16 eventos posibles fue satisfactoria.

En la pantalla de Función Principal se observa que al lado izquierdo se indica el pasajero correspondiente para cada línea; en el centro se muestra “LIBRE” si no

hay pasajero a bordo o “A PAGAR” en caso de haberlo. Y finalmente a la derecha se muestra el precio en C\$ a medida que se avanza sobre la carrera.

Funcionalidad en Menú Terminar Carrera

Para ingresar al Menú Terminar Carrera desde el Estado Función Principal se debe presionar el botón STOP. En este menú existen 5 opciones de navegación las cuales están presididas por un cursor, como se puede observar en las fotografías de la figura 4.9.



Ilustración 43- Pantalla Menú Terminar Carrera

El cursor es controlado por los botones direccionales (arriba y abajo), y se puede acceder a las opciones de navegación a través del botón OK. Si se selecciona la opción 5 (Salir) el prototipo regresa al Estado Función Principal. Las carreras en curso están indicadas por una viñeta en la parte derecha de cada uno de los pasajeros, como se puede observar en la figura anterior junto al Pasajero 1 y Pasajero 2. Si se selecciona una opción que no tenga viñeta el prototipo regresa al Estado Función Principal. Si se selecciona alguna de las carreras en curso se ingresa al submenú Imprimir Recibo como se muestra a continuación:

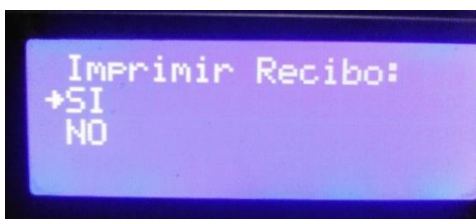


Ilustración 44- Pantalla Menú Terminar Carrera/Imprimir Recibo

Si el usuario selecciona la opción 1 (SI) el sistema procede a imprimir el recibo y a su vez muestra en pantalla en letra grande el precio de la carrera. Si es la opción 2 (NO) que fue seleccionada el sistema muestra en pantalla en letra grande el precio de la carrera y permanece en esta pantalla hasta que se presione nuevamente STOP, aunque cabe destacar que luego de 30 segundos en esta pantalla el buzzer empieza a emitir pitidos intermitentes como advertencia de que aún se permanece en esta pantalla.



Ilustración 45- Pantalla Menú Terminar Carrera/A Pagar

6.2 Pruebas de Impresión

El equipo está programado para imprimir dos tipos de documentos si se añade el módulo externo de la impresora, que son ticket identificativo y recibo. El ticket identificativo muestra el número de pasajero a como se muestra a continuación:

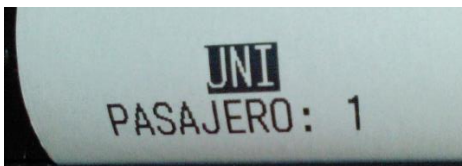


Ilustración 46- Prueba de Impresión/Ticket Identificativo

El recibo indica el pasajero que está finalizando su carrera, fecha y hora de emisión del recibo, tipo de tarifa (diurna o nocturna), distancia recorrida, y finalmente el valor a pagar. El resultado final puede apreciarse en la siguiente figura:



Ilustración 47- Prueba de Impresión/Recibo

Como se puede observar ambas figuras, las pruebas de impresión han culminado con éxito.

6.3 Pruebas De Distancia

Para realizar las pruebas de distancia se necesitó de un vehículo con conector de mechero, las pruebas se realizaron en la carretera nueva a León, exactamente desde el kilómetro 12 hasta el kilómetro 7 tomando como punto de referencia los hitos kilométricos, 5 KM en total. Se realizaron 6 pruebas con velocidades comprendidas entre 20 y 80 Km/h obteniendo los siguientes resultados:

No. Prueba	Distancia Registrada	Error en %
Prueba 1	5.00 KM	0.0%
Prueba 2	5.00 KM	0.0%
Prueba 3	5.01 KM	0.2%
Prueba 4	4.98 KM	0.4%
Prueba 5	5.00 KM	0.0%
Prueba 6	5.01 KM	0.2%

Error Promedio en %	0.13%
Error Promedio en KM	0.0065 KM

Tabla 10- Pruebas de Tiempo de Conexión GPS en Carretera Abierta

Basados en las pruebas realizadas referentes a distancia es notable la precisión del taxímetro digital multipasajeros. Con un error en promedio de 0.13% su precisión se estima en 99.87%, bastante acertado para el propósito para el cual fue diseñado. Hablando del error en KM se puede observar que es de 0.0065 KM o su equivalente en metros de 6.5 metros; muy congruente con la realidad. Hay que destacar que estas pruebas fueron realizadas en carretera abierta como normalmente circularía un vehículo al momento de laborar.

6.4 Pruebas de Costos por Carrera

Para las pruebas de costos por carreras se realizaron 3 recorridos hacia puntos comunes dentro de la capital. El primer recorrido fue desde Linda Vista hasta la Universidad Centroamericana UCA (8.2 Km). El segundo recorrido fue desde el Km 7 de la carretera Sur hasta la Universidad Centroamericana UCA (4.5 Km). Y finalmente el tercer recorrido fue desde el centro comercial Plaza Inter hasta el Puerto Salvador Allende (2.1 Km). Estos tres recorridos fueron realizados simulando 1 pasajero, 2 pasajeros, 3 pasajeros y 4 pasajeros a bordo logrando un total de 12 carreras distintas.

- Recorrido Linda Vista-UCA:

Recorrido Linda Vista-UCA		
<i>Pasajeros a bordo</i>	<i>Precio por Persona</i>	<i>Distancia Recorrida</i>
1	C\$ 70	8.2 Km
2	C\$ 45	8.2 Km
3	C\$ 35	8.2 Km
4	C\$ 30	8.2 Km

Tabla 11- Resultados Recorrido Linda Vista-UCA

En el recorrido Linda Vista-UCA simulando un pasajero a bordo el sistema facturó un precio de C\$70. Este valor representa el costo mínimo suficiente para cubrir los gastos de dicha carrera.

Se realizó el mismo trayecto simulando dos pasajeros a bordo donde el precio a pagar por persona se redujo a C\$45, que representa un ahorro del 35.71% por pasajero respecto al costo de C\$70 en caso de que estas personas realizaran la misma carrera de manera individual.

Se procedió a realizar una tercera prueba en el mismo trayecto simulando tres personas a bordo para lo cual el sistema facturó un total de C\$35 por persona, que representa un ahorro del 50% por pasajero en comparación a lo que el sistema factura en una carrera de un solo pasajero (C\$70).

Sobre el mismo trayecto finalmente se simuló un escenario de 4 personas que parten del mismo origen hacia el mismo final. Los resultados para el precio por persona fueron de C\$30 en el cual cada pasajero se ahorra un total de C\$40 que representa el 57.14% respecto al precio de C\$70, valor que representa el costo de la misma carrera si se realiza de manera individual.

- Recorrido UCA-7Sur:

Recorrido UCA-7Sur		
<i>Pasajeros A Bordo</i>	<i>Precio por Persona</i>	<i>Distancia Recorrida</i>
1	C\$ 45	4.5 Km
2	C\$ 35	4.5 Km
3	C\$ 30	4.5 Km

4	C\$ 25	4.5 Km
---	--------	--------

Tabla 12- Resultados Recorrido UCA-7Sur

En la segunda ruta UCA-7Sur simulando un pasajero a bordo el sistema facturó un precio de C\$45.

Se realizó el mismo trayecto UCA-7Sur simulando dos pasajeros a bordo obteniendo un precio a pagar por persona de C\$35, representando un ahorro del 22.22% por pasajero respecto al costo de C\$45 en caso de que estas personas realizaran la misma carrera de manera individual.

Luego se realizó la tercera prueba en el mismo trayecto simulando tres personas a bordo para lo cual el sistema facturó un total de C\$30 por persona, que representa un ahorro del 33.33% por pasajero en comparación a lo que el sistema factura en una carrera de un solo pasajero (C\$45).

Finalmente se simuló un escenario de 4 personas compartiendo simultáneamente el vehículo para este mismo recorrido. El precio por persona fue de C\$25 en el cual cada pasajero se ahorra un 44.44% respecto al precio de C\$45, valor que representa el costo de la misma carrera si estas 4 personas viajasen por separado.

- Recorrido Plaza Inter-Salvador Allende:

Recorrido Plaza Inter-Salvador Allende		
<i>Pasajeros A Bordo</i>	<i>Precio por Persona</i>	<i>Distancia Recorrida</i>
1	C\$ 30	2.1 Km
2	C\$ 25	2.1 Km
3	C\$ 25	2.1 Km

4	C\$ 25	2.1 Km
---	--------	--------

Tabla 13- Resultados Plaza Inter-Salvador Allende

En el recorrido Plaza Inter-Salvador Allende simulando un pasajero a bordo el sistema facturó un precio de C\$30. Este valor representa el costo mínimo suficiente para cubrir los gastos de dicha carrera.

Se realizó el mismo trayecto simulando dos pasajeros a bordo donde el precio a pagar por persona se redujo a C\$25, que representa un ahorro del 16.67% por pasajero respecto al costo de C\$30 en caso de que estas personas realizaran la misma carrera de manera individual.

Se procedió a realizar una tercera prueba en el mismo trayecto simulando tres personas a bordo para lo cual el sistema facturó un total de C\$25 por persona, que representa el mismo ahorro que para el caso anterior del 16.67% por pasajero en comparación a los C\$30 que el sistema factura en una carrera de un solo pasajero.

Sobre el mismo trayecto finalmente se simuló el escenario de 4 pasajeros a bordo. Los resultados para el precio por persona fueron de C\$25 en el cual cada pasajero se ahorra un 16.67% respecto al costo de C\$30 en caso de que estas personas realizaran la misma carrera de manera individual, de manera similar a la posible situación de 2 y de 3 pasajeros a bordo.

Basados en los datos obtenidos se puede deducir que el precio a pagar disminuye según el número de pasajeros a bordo. También, para rutas de mayor distancia con un mayor número de pasajeros, se obtiene un mayor ahorro en comparación a rutas más cortas con la misma cantidad de pasajeros donde el ahorro no es muy significativo.

Ahora bien, hablando sobre el análisis del impacto de estos resultados sobre los cadetes, las ganancias del conductor están en función de la cantidad de pasajeros que transporten en un turno oficial y no en función de la distancia del recorrido. Es decir que a mayor número de pasajeros ambas partes son beneficiadas, tanto usuarios como cadetes. Como se mencionó en el apartado “Cálculo de Tarifa”, el cadete necesita realizar 26 carreras al día como meta. Si transporta un solo pasajero sobre determinada ruta, necesitaría 25 carreras más. Pero si transporta 4 pasajeros sobre la misma ruta, necesitaría 22 carreras más en vez de 25.

6.5 Análisis Económico

El siguiente análisis refleja el costo de un taxímetro según el diseño propuesto. Se realizaron 2 tablas, una para la tarjeta principal y otra para la de comunicación USB. El valor de la tarjeta principal se detalla a continuación.

TARJETA PRINCIPAL				
Ítems	Descripción	Cantidad	Precio (U\$)	Total (U\$)
1	DIODE GEN PURP 1KV 1A SMA	1	0.24	0.24
2	CAP ALUM 10UF 20% 50V RADIAL	1	0.3	0.3
3	CAP TANT 10UF 25V 10% RADIAL	1	0.5	0.5
4	IC REG LDO 5V 3A TO220-3	1	2.06	2.06
5	FUSE GLASS 600MA 250VAC 125VDC	1	0.94	0.94
6	BERGSTIK II .100" SR STRAIGHT	2	0.27	0.54
7	CONN IC DIP SOCKET 28POS TIN	1	0.77	0.77
8	IC MCU 8BIT 32KB FLASH 28DIP	1	3.74	3.74
10	CRYSTAL 16MHZ 20PF THRU	1	1.12	1.12
11	CONN HEADER VERT DUAL 6POS GOLD	1	0.42	0.42

12	CAP CER 22PF 50V 5% RADIAL	2	0.34	0.68
14	CAP CER 0.1UF 50V 10% X7R 1206	2	0.1	0.2
19	RES 1K OHM 1/4W 5% AXIAL	4	0.1	0.4
20	RES 10K OHM 1/4W 5% CARBON FILM	6	0.1	0.6
24	TRIMMER 10K OHM 0.5W PC PIN	1	1.6	1.6
25	TRANS NPN 30V 0.1A TO-92	1	0.22	0.22
26	LCD MOD CHAR 20X4 WHT TRANSMISS	1	4.21	4.21
27	SWITCH TACTILE SPST-NO 0.05A 24V	6	0.15	0.9
28	BUZZER PIEZO 4KHZ 12.2MM PC MNT	1	0.72	0.72
29	LED RGB	1	0.5	0.5
30	GPS MODULE	1	39.95	39.95
31	PCB 100mmx60mm	1	6	6
TOTAL				66.61

Tabla 14- Costo Total PCB Principal

Este análisis contempla el desarrollo de una única placa, pero es importante remarcar que si la producción fuese en masa, los precios se reducirían de manera considerable. Los precios no incluyen detalles como precio por empaquetado y costos de envío. Se excluye el valor de la impresora ya que es un módulo externo, agregable a criterio del usuario final. La impresora por sí sola tiene un costo de 49.95 dólares.

A continuación se detalla el valor de la tarjeta para la comunicación USB.

TARJETA DE COMUNICACIÓN USB				
Ítems	Descripción	Cantidad	Precio (U\$)	Total (U\$)
1	BERGSTIK II .100" SR STRAIGHT	2	0.27	0.54
2	IC MCU 8BIT 8KB FLASH 32TQFP	1	4.38	4.38

3	CRYSTAL 16MHZ 20PF THRU	1	1.12	1.12
4	CONN HEADER VERT DUAL 6POS GOLD	1	0.42	0.42
5	CAP CER 22PF 50V 5% RADIAL	2	0.34	0.68
6	CAP CER 1UF 50V RADIAL	1	0.55	0.55
7	CAP CER 0.1UF 50V 10% X7R 1206	3	0.1	0.3
11	RES 1K OHM 1/4W 5% AXIAL	4	0.1	0.4
12	RES 10K OHM 1/4W 5% CARBON FILM	1	0.1	0.1
13	RES 22 OHM 1/4W 5% CARBON FILM	2	0.1	0.2
14	PCB 80mm x 30mm	1	4.84	4.84
15	CONN USB RECEPT R/A TYPE B 4POS	1	1.92	1.92
TOTAL				15.45

Tabla 15- Costo Total PCB de Programación USB

En este apartado es importante destacar que este valor de \$15.45 es por realizar una única tarjeta. La producción en masa reduciría su costo y agregado a esto se encuentra el hecho de que una sola tarjeta puede ser utilizada para programar todas las tarjetas principales de las que disponga el usuario final, una a la vez.

7 CONCLUSIONES

Inicialmente para el diseño y construcción del taxímetro digital multipasajeros se identificaron los principales requerimientos y limitantes, que son: la inexistencia de un modelo tarifario en el sector taxi, sistema de taxis ruleteros (múltiples carreras simultáneas) y la mínima irrupción en el vehículo al momento de la instalación.

El valor del dispositivo es de \$66.61 que hace del sistema una opción bastante asequible en comparación a los taxímetros comunes en el mercado que no ofrecen la capacidad de carreras simultáneas, ya que los costos de los taxímetros que se han promocionado al país ascienden a los \$200. Si se desea incluir impresora, el costo total del sistema es de \$116.51, aún muy por debajo de los precios ofertados.

La simulación y construcción del hardware del taxímetro se desarrolló con Altium Designer, software profesional del cual se lograron obtener nuevos conocimientos y que facilitaron la exportación del diseño hacia otros programas como SketchUp para el mecanizado del empaquetado.

Se adaptó un módulo GPS al sistema, que brinda datos con una precisión de hasta 99.87% en la medición de distancia; sin embargo el taxímetro es adaptable con otros módulos GPS que pueden representar un menor costo para el dispositivo. También se anexó una impresora de tickets y de recibos como módulo opcional donde el costo del dispositivo disminuye considerablemente sin dicha impresora.

Para el desarrollo de la programación del microcontrolador se utilizó la herramienta Atmel Studio 6.2, con un lenguaje C para AVR que permite la optimización sin tantas complicaciones al momento de redactar las líneas de

código y que cuenta con una amplia variedad de librerías que facilitan el desarrollo del firmware.

Las medidas del prototipo de 150 mm x 100 mm x 60 mm permiten utilizar un espacio reducido dentro del vehículo, lo que facilita su instalación sin afectar el resto del sistema del auto, ni la visibilidad, ni la comodidad del conductor o de los pasajeros.

En conclusión, fue posible construir un taxímetro con la capacidad de facturar cuatro carreras de manera simultánea y justa gracias al GPS y al microcontrolador basados en datos veraces proporcionados por FENICOOTAXI e IRTRAMMA, siendo este un aparato de fácil uso para el cadete, con un número reducido de botones de mando, con mensajes de alerta y de correcto funcionamiento gracias a los LEDs, con capacidad de acople con una impresora de tickets y recibos, con un pequeño tamaño que permite su fácil instalación y únicamente necesitando del vehículo la alimentación eléctrica sin irrumpir en su mecanismo. Y algo remarcable: todas estas ventajas a un muy bajo costo.

8 RECOMENDACIONES

Las siguientes son recomendaciones para la mejora al momento del uso del taxímetro digital multipasajeros y para un tiempo posterior.

Ayuda Visual: en caso de no desear el módulo de impresión de tickets/recibos de parte del dueño del vehículo y utilizar las fichas de colores identificativas, se recomienda tener una página impresa indicando que color de ficha corresponde a que pasajero para evitar cualquier tipo de confusión al momento de prestar el servicio a más de un pasajero a la vez. Colocar esta página impresa en un lugar visible tanto para el conductor como para los pasajeros para que haya mejor comprensión de ambas partes.

Recepción del GPS: la antena del GPS debe colocarse en un lugar donde tenga la menor cantidad posible de obstáculos que dificulten la línea de vista hacia los satélites, ya que necesita una señal bastante fuerte para la recepción de datos y para poder inicializar el posicionamiento.

Segunda Pantalla en la Parte Posterior del Vehículo: la pantalla LCD de la que dispone el dispositivo puede no ser muy clara para las personas que vayan ubicadas en los asientos posteriores del automóvil. Por tanto, para ellos poder conocer su tarifa es mejor que posean de una pantalla aparte ubicada cerca de ellos.

Capacitación de los Cadetes: el sistema ha sido simplificado para facilitar su comprensión, sin embargo para un nuevo conductor puede resultar complejo trabajar con este sistema por lo que no está habituado. Se recomienda que antes de utilizar este taxímetro se lleve a cabo la correspondiente capacitación y la continua enseñanza en caso de haber algún cambio.

Levantar Datos Estadísticos: se recomienda utilizar este taxímetro como otra fuente de información para datos como distancias recorridas con pasajeros, sin pasajeros, en total, etc. Esto servirá como apoyo y facilitación del trabajo que se realiza por las distintas cooperativas para llevar el control sobre sus unidades.

Central de Monitoreo: aprovechando las capacidades que el GPS proporciona al sistema, es posible que en un futuro se le añada la función GSM para poder recibir los datos en tiempo real en una central de monitoreo para llevar registro de todo lo que hacen los cadetes al volante. Esto serviría para modernizar aún más el servicio y para que las cooperativas conozcan la actividad que lleva a cabo el sector taxi.

9 BIBLIOGRAFIA

1. Miguel Antonio Cuasapaz Chamorro; Construcción e implementación de un prototipo de un taxímetro digital con impresora facturadora utilizando tecnología GPS y el desarrollo de un firmware en un microcontrolador, disponible en: <http://bibdigital.epn.edu.ec/bitstream/15000/6210/1/CD-4832.pdf>
2. Becker Andrés Sanguano Fierro; Estudio, diseño e implementación de un sistema prototipo de control vía mensaje de texto en un taxímetro, disponible en: <http://186.42.96.211:8080/jspui/bitstream/123456789/898/1/bsanguano-ecalvache.pdf>
3. Guía de cuestiones a tener presentes en la definición de tarifas en el sector del taxi, Centro Español de Metrología, Centro Español de Metrología, disponible en: http://www.cem.es/sites/default/files/files/Guia_Tarifas_Taxi.pdf
4. Luis Leonardo Rivera Abaunza; Taxímetro Digital en VHDL, disponible en: <http://www.monografias.com/trabajos-pdf2/taximetro-digital-vhdl/taximetro-digital-vhdl.pdf>
5. Jorge R. Rey; El Sistema de Posicionamiento Global – GPS, Universidad de Florida, disponible en: edis.ifas.ufl.edu/pdf/IN/IN65700.pdf
6. Ing. Jairo Noel García Pérez, GPS Global Positioning System, Sistemas de Posicionamiento Global; Instituto de Ciencias Básicas e Ingeniería, Universidad Autónoma del Estado de Hidalgo, Monografía disponible en: dgsa.uaeh.edu.mx:8080/bibliotecadigital/bitstream/231114/211/1/Gps%20global.pdf
7. Carlos Canto Q.; Microcontroladores, Facultad de Ciencias UASLP, disponible en: http://galia.fc.uaslp.mx/~cantocar/microcontroladores/SLIDES_8051_PDF/2_INTROD.PDF

8. Miguel Torres Torreti; Tutorial Microcontroladores Pic - Iniciación Rápida -, disponible en: http://web.ing.puc.cl/~mtorrest/downloads/pic/tutorial_pic.pdf
9. ¿Qué es un Microcontrolador?, Guía del estudiante, disponible en: www.parallax.com/sites/default/files/downloads/28123-Whats-A-Microcontroller-Espanol-v3.0.pdf
10. Edelsys Hernández Meléndrez; Cómo escribir una tesis, disponible en: biblioteca.ucv.cl/site/servicios/documentos/como_escribir_tesis.pdf
11. Modelo de Regulación Económica del Servicio Público de Transporte Remunerado de Personas, Modalidad Taxis; Costa Rica.
12. Ley 524: Ley General de Transporte Terrestre; publicado en La Gaceta No.72 en Managua, Nicaragua, 14 de Abril de 2005.
13. Ampliación bajo el Decreto No. 42-2005 a la Ley 524: Ley General de Transporte Terrestre; publicado en La Gaceta No.113 en Managua, Nicaragua, 16 de Junio de 2005.
14. Atmel. ATmega328P Datasheet, Atmel Corporation, 2009.
15. Atmel. ATmega8U2 Datasheet, Atmel Corporation, 2010.
16. Tarifa Nocturna de Taxis Ruleteros de la Ciudad de Managua; Acuerdo Ministerial No. 108; Managua, Nicaragua, 07 de Agosto de 1984.

10 REFERENCIAS

1. www.laprensa.com.ni/2014/12/07/economia/1437051-gasolinas-caen-mas-del-25-gasolinas-caen-mas-del-25
2. m.diccionario.motorgiga.com/diccionario/taxímetro-definicion-significado/gmx-niv15-con195692.htm
3. Acuerdo Ministerial No. 108; Tarifa Nocturna de Taxis Ruleteros de la Ciudad de Managua; Managua, Nicaragua, 07 de Agosto de 1984.
4. Ley 524: Ley General de Transporte Terrestre; Managua, Nicaragua, 16 de Junio de 2005.
5. m.diccionario.motorgiga.com/diccionario/taxímetro-definicion-significado/gmx-niv15-con195692.htm
6. Guía de cuestiones a tener presentes en la definición de tarifas en el sector del taxi, Centro Español de Metrología,
http://www.cem.es/sites/default/files/files/Guia_Tarifas_Taxi.pdf
7. <http://www.gps.gov/systems/gps/>
8. <http://edis.ifas.ufl.edu/pdf/IN/IN65700.pdf>
9. NMEA: Manual de Referencia, disponible en:
<https://www.sparkfun.com/datasheets/GPS/NMEA%20Reference%20Manual1.pdf>
10. <http://www.aquihayapuntes.com/indice-practicas-pic-en-c/comunicacion-usb-pic18f4550-utilizando-la-clase-cdc.html>
11. http://galia.fc.uaslp.mx/~cantocar/microcontroladores/SLIDES_8051_PDF/2_INTROD.PDF

12. Datasheet ATmega328P

13. Datasheet ATmega8U2

14. Modelo de Regulación Económica del Servicio Público de Transporte
Remunerado de Personas, Modalidad Taxi

ANEXOS

ANEXO 1 – CODIGO FUENTE

/*****

*

* Name: Taxímetro Digital Multipasajeros con GPS e Impresora de Recibos

* Autor: Kelvin Olivares y Edgard Avendaño

* Web: http://cacionelectronics.com

* Version: 2.6 *****

* Date: 2016/05/11

/

/*****/

/***** LIBRERIAS Y OBJETOS *****/

/*****/

#include <Time.h>

#include <TinyGPS++.h> //Librería para el manejo del GPS.

#include <SoftwareSerial.h> //Librería para el manejo de los puertos virtuales
TTL.

#include <Wire.h>

#include <Adafruit_Thermal.h> //Librería para el manejo de la impresora.

#include <LiquidCrystal.h> //Librería para el manejo de la pantalla LCD.

#include <avr/sleep.h> //Librería para el manejo de ahorro de energía en
AVR.

#include <avr/pgmspace.h> //Librería para almacenar información en la
memoria de programa.

#include <EEPROM.h> //Librería para el manejo de la memoria
EEPROM.

```
#define Led_Azul   PC3
#define Led_Rojo   PC4
#define Led_Verde  PC5
#define buzzer     4
#define led_Lcd    7
```

```
static const int RX1 = 6, TX1 = 5;
LiquidCrystal lcd(13, 12, 11, 10, 9, 8);
TinyGPSPlus gps;
SoftwareSerial Gps(RX1,TX1);
Adafruit_Thermal printer(&Serial);
/*****
/***** VARIABLES GLOBALES *****/
/*****/
```

```
byte menuPosition      = 1; // Posicion del cursor en el Menú
const byte menuOptions  = 6; // Numero de opciones de configuracion del
menu principal
const byte rowsLCD      = 4; // Numero de filas del LCD
const int UTC_offset    = -6; // Zona Horaria UTC en Managua, Nicaragua
time_t prevDisplay      = 0; // Count for when time last displayed
float distance;          // Variables que guardan valores de distancias
float d1, d2, d3, d4;    // distancias totales.
byte btnDown;           // Botón para bajar el cursor en el LCD
byte btnUp;             // Botón para subir el cursor en el LCD
byte btnCenter;         // Botón para acceder a la opción en el Menú
byte Estado;            // Cantidad de Pasajeros a bordo
boolean Salir   = false; // Variable de control del Menu principal
boolean Exit    = false; // Variable de control de la máquina de estados
```

```

volatile boolean BNP = HIGH;      // Variable para controlar si se presionó el
btnNewP
volatile boolean BSTOP = HIGH;    // Variable para controlar si se presionó el
btnStop
unsigned long tiempo;
float distancia1 = 0;
float distancia2 = 0;
float distancia3 = 0;
float distancia4 = 0;
uint8_t Tarifa = 1;
float lat1;
float lon1;
float lat2;
float lon2;
int vel;

/* MENU DE SALIDA */
const char *nMENU[] = {
    "Terminar Carrera: ",
    "Pasajero 1      ",
    "Pasajero 2      ",
    "Pasajero 3      ",
    "Pasajero 4      ",
    "Salir           "
};

/***** INICIALIZACION DE LAS VARIABLES *****/

void setup()

```

```

{
  // Configura los puertos del ATMEGA 328P: B, C, D
  DDRB = DDRB | B00111111; // Puerto B como Salida
  DDRC = DDRC | B111000; // Configurando el puerto C; C0,C1,C2 como
  entrada,C3 AZUL ,C4 ROJO, C5 GREEN salidas.
  PORTC = (0 << Led_Verde) | (1 << Led_Rojo) | (1 << Led_Azul); //

  // Inicia el LCD:
  lcd.begin(20,4);
  pinMode(led_lcd,OUTPUT);
  digitalWrite(led_lcd,HIGH);
  pinMode(buzzer,OUTPUT);
  digitalWrite(buzzer,LOW);

  // Inicia y configura los puertos virtuales UART:
  Serial.begin(19200); // setup the software serial pins
  Gps.begin(4800); // setup the software serial pins
  printer.begin();

  /// Habilita las interrupciones INT0 e INT1:
  attachInterrupt( 1, Nuevo_Pasajero, FALLING);
  attachInterrupt( 0, Detener_Carrera, FALLING);
  set_sleep_mode(SLEEP_MODE_PWR_DOWN); //
}

/*****/
/***** FUNCION CICLICA *****/
/*****/

void loop()
{
  while (Gps.available() > 0)

```

```

gps.encode(Gps.read());

if(gps.speed.isValid())
{
    lcd.setCursor(5,0);
    lcd.print(F("TAXIMETRO"));
    lcd.setCursor(5,1);
    lcd.print(F(" LIBRE "));
    vel = gps.speed.kmph();
    lcd.setCursor(11,3);
    lcd.print("Km/Hr");

    if (vel < 10)
    {
        lcd.setCursor(7,3);
        lcd.print(" ");
        lcd.setCursor(9,3);
        lcd.print(vel);
    }

    else if (vel >= 10 && vel < 100)
    {
        lcd.setCursor(7,3);
        lcd.print(" ");
        lcd.setCursor(8,3);
        lcd.print(vel);
    }

    else if (vel >= 100)
    {
        lcd.setCursor(7,3);
        lcd.print(vel);
    }
}

```

```

    }

    if (timeStatus() != timeNotSet)
    {
        if (now() != prevDisplay)
        {
            prevDisplay = now();
        }
    }
}

else
{
    lcd.setCursor(5,1);
    lcd.print(F("Cargando"));
    lcd.setCursor(2,2);
    lcd.print(F("Configuracion..."));
}

if(BNP == LOW) // Se presionó el botón NP?
{
    if(gps.location.isValid()) // El GPS está enganchado
    {
        Salir = false;
        Agregar_Pasajero();
        Funcion_Principal();
    }
    else // EL GPS no está enganchado
    {
        lcd.setCursor(5,1);
        lcd.print(F("Cargando"));
    }
}

```

```

        lcd.setCursor(2,2);
        lcd.print(F("Configuracion..."));
        BNP = LOW;
    }
}

if (millis() > 5000 && gps.charsProcessed() < 10) // GPS desconectado
{
    PORTC = (1 << Led_Verde) | (0 << Led_Rojo) | (1 << Led_Azul); // Led
    ROJO encendido
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("GPS NO DETECTADO");
    while(true);
}
}

/* Función que simula una máquina de estados.
 * Parámetro Estado: Indica qué pasajeros estan abordo, 0: Indica Ausencia de
pasajero; 1: Indica Existencia de pasajero;
 * Parámetro D1   : Distancia parcial recorrida por el Pasajero 1.
 * Parámetro D2   : Distancia parcial recorrida por el Pasajero 2.
 * Parámetro D3   : Distancia parcial recorrida por el Pasajero 3.
 * Parámetro D4   : Distancia parcial recorrida por el Pasajero 4.
 */

void Funcion_Principal()
{
    lcd.clear();
    d1 = 0; d2 = 0; d3 = 0; d4 = 0;
    distance = 0;

```



```
BNP = HIGH;  
BSTOP = HIGH;
```

```
while(!Exit)  
{  
    lcd.setCursor(0,0); lcd.print("P1.");  
    lcd.setCursor(0,1); lcd.print("P2.");  
    lcd.setCursor(0,2); lcd.print("P3.");  
    lcd.setCursor(0,3); lcd.print("P4.");  
  
    if (BNP == LOW && Estado != B1111) //  
    {  
        Salir = false;  
        Agregar_Pasajero(); // dirigimos al menu agregar pasajero  
    }  
    Sensor_Odometro();  
  
    //Pasajero 1 ON, Pasajero 2, 3, 4 OFF  
    if (Estado == B0001)  
    {  
        d1 += distance;  
        d2 = 0; d3 = 0; d4 = 0;  
        distancia1 += distance;  
    }  
  
    //Pasajero 2 ON, Pasajero 1, 3, 4 OFF  
    if(Estado == B0010)  
    {  
        d2 += distance;  
        d1 = 0; d3 = 0; d4 = 0;  
        distancia2 += distance;
```

```
}
```

```
//Pasajero 1, 2 ON, Pasajero 3, 4 OFF
```

```
if(Estado == B0011)
```

```
{
```

```
    d1 += distance;
```

```
    d2 += distance;
```

```
    d3 = 0; d4 = 0;
```

```
    distancia1 += distance/2;
```

```
    distancia2 += distance/2;
```

```
}
```

```
//Pasajero 3 ON, Pasajero 1, 2, 4 OFF
```

```
if (Estado == B0100)
```

```
{
```

```
    d3 += distance;
```

```
    d1 = 0; d2 = 0; d4 = 0;
```

```
    distancia3 += distance;
```

```
}
```

```
//Pasajero 1, 3 ON, Pasajero 2, 4 OFF
```

```
if(Estado == B0101)
```

```
{
```

```
    d1 += distance;
```

```
    d3 += distance;
```

```
    d2 = 0; d4 = 0;
```

```
    distancia1 += distance/2;
```

```
    distancia3 += distance/2;
```

```
}
```

```
//Pasajero 2, 3 ON, Pasajero 1, 4 OFF
```

```
if(Estado == B0110)
{
    d2 += distance;
    d3 += distance;
    d1 = 0; d4 = 0;
    distancia2 += distance/2;
    distancia3 += distance/2;
}
```

//Pasajero 1, 2, 3 ON, Pasajero 4 OFF

```
if(Estado == B0111)
{
    d1 += distance;
    d2 += distance;
    d3 += distance;
    d4 = 0;
    distancia1 += distance/3;
    distancia2 += distance/3;
    distancia3 += distance/3;
}
```

//Pasajero 4 ON, Pasajero 1, 2, 3 OFF

```
if(Estado == B1000)
{
    d4 += distance;
    d1 = 0; d2 = 0; d3 = 0;
    distancia4 += distance;
}
```

//Pasajero 1, 4 ON, Pasajero 2, 3 OFF

```
if(Estado == B1001)
```

```
{
    d1 += distance;
    d4 += distance;
    d2 = 0; d3 = 0;
    distancia1 += distance/2;
    distancia4 += distance/2;
}

//Pasajero 2, 4 ON, Pasajero 1, 3 OFF
if(Estado == B1010)
{
    d1 = 0;
    d2 += distance;
    d3 = 0;
    d4 += distance;
    distancia2 += distance/2;
    distancia4 += distance/2;
}

//Pasajero 1, 2, 4 ON, Pasajero 3 OFF
if(Estado == B1011)
{
    d1 += distance;
    d2 += distance;
    d3 = 0;
    d4 += distance;
    distancia1 += distance/3;
    distancia2 += distance/3;
    distancia4 += distance/3;
}
```

```
//Pasajero 3, 4 ON, Pasajero 1, 2 OFF
```

```
if (Estado == B1100)
```

```
{
```

```
    d1 = 0; d2 = 0;
```

```
    d3 += distance;
```

```
    d4 += distance;
```

```
    distancia3 += distance/2;
```

```
    distancia4 += distance/2;
```

```
}
```

```
//Pasajero 1, 3, 4 ON, Pasajero 2 OFF
```

```
if (Estado == B1101)
```

```
{
```

```
    d1 += distance;
```

```
    d2 = 0;
```

```
    d3 += distance;
```

```
    d4 += distance;
```

```
    distancia1 += distance/3;
```

```
    distancia3 += distance/3;
```

```
    distancia4 += distance/3;
```

```
}
```

```
//Pasajero 2, 3, 4 ON, Pasajero 1 OFF
```

```
if (Estado == B1110)
```

```
{
```

```
    d1 = 0;
```

```
    d2 += distance;
```

```
    d3 += distance;
```

```
    d4 += distance;
```

```
    distancia2 += distance/3;
```

```
    distancia3 += distance/3;
```

```
    distancia4 += distance/3;
}
```

```
//Pasajero 1, 2, 3, 4 ON
```

```
if (Estado == B1111)
{
    d1 += distance;
    d2 += distance;
    d3 += distance;
    d4 += distance;
    distancia1 += distance/4;
    distancia2 += distance/4;
    distancia3 += distance/4;
    distancia4 += distance/4;
}
```

```
if((Estado & B0001) == B0001)
{
    lcd.setCursor(4,0); lcd.print("A PAGAR:");
    lcd.setCursor(13,0); lcd.print(Calcular_Precio(distancia1));
    lcd.setCursor(18,0); lcd.print(F("C$"));
}
else
{
    lcd.setCursor(7,0); lcd.print("LIBRE");
}
```

```
if((Estado & B0010) == B0010)
{
    lcd.setCursor(4,1); lcd.print("A PAGAR:");
    lcd.setCursor(13,1); lcd.print(Calcular_Precio(distancia2));
}
```

```

        lcd.setCursor(18,1); lcd.print(F("C$"));
    }
    else
    {
        lcd.setCursor(7,1); lcd.print("LIBRE");
    }

    if((Estado & B0100) == B0100)
    {
        lcd.setCursor(4,2); lcd.print("A PAGAR:");
        lcd.setCursor(13,2); lcd.print(Calcular_Precio(distancia3));
        lcd.setCursor(18,2); lcd.print(F("C$"));
    }
    else
    {
        lcd.setCursor(7,2); lcd.print("LIBRE");
    }

    if((Estado & B1000) == B1000)
    {
        lcd.setCursor(4,3); lcd.print("A PAGAR:");
        lcd.setCursor(13,3); lcd.print(Calcular_Precio(distancia4));
        lcd.setCursor(18,3); lcd.print(F("C$"));
    }
    else
    {
        lcd.setCursor(7,3); lcd.print("LIBRE");
    }

    if(BSTOP == LOW) //Se presionó el btn STOP??
    {

```

```

        Salir = false;
        Terminar_Carrera(); // Menu terminar carrera
    }

    //Pasajero 1, 2, 3, 4 OFF
    if(Estado == B0000)
    {
        d1 = 0; d2 = 0; d3 = 0; d4 = 0;
        menuPosition = 1;
        Salir = true;
        Exit = true;
    }
}

BNP = HIGH;
lcd.clear();
}

```

/* Muestra el Menú Principal del Taxímetro

* Parámetro Estado: Indica qué pasajeros están abordo, 0: Indica Ausencia de pasajero; 1: Indica Existencia de pasajero;

* Parámetro d1 : Distancia parcial recorrida por el Pasajero 1

* Parámetro d2 : Distancia parcial recorrida por el Pasajero 2

* Parámetro d3 : Distancia parcial recorrida por el Pasajero 3

* Parámetro d4 : Distancia parcial recorrida por el Pasajero 4

*/

```
void Agregar_Pasajero()
```

```

{
    boolean estado = false;
    menuPosition = 1;
    lcd.clear();

```



```

tiempo = millis();

while(!Salir)
{
    readButtons();
    if (!btnDown && menuPosition == 1) // Se presionó el boton DOWN?
    {
        tiempo = millis();
        menuPosition=2;
    }
    else if (!btnUp && menuPosition == 2) //Se presionó el boton UP?
    {
        tiempo = millis();
        menuPosition=1;
    }
    else if(!btnCenter ) //Se presionó el boton OK?
    {
        switch( menuPosition )
        {
            case 1: // Se seleccionó la opcion 1 (SI)
                if((Estado & B0001) == B0000 && estado == false) // Pasajero 1
                    agregado
                {
                    Imprimir_Ticket(1);
                    estado = true;
                    Estado = (Estado | B0001);
                    Exit = false; Salir = true;
                }
                if((Estado & B0010) == B0000 && estado == false) // Pasajero 2
                    agregado
                {

```

```

        Imprimir_Ticket(2);
        estado = true;
        Estado = (Estado | B0010);
        Exit = false; Salir = true;
    }
    if((Estado & B0100) == B0000 && estado == false) // Pasajero 3
agregado
    {
        Imprimir_Ticket(3);
        estado = true;
        Estado = (Estado | B0100);
        Exit = false; Salir = true;
    }
    if((Estado & B1000) == B0000 && estado == false) // Pasajero 4
agregado
    {
        Imprimir_Ticket(4);
        estado = true;
        Estado = (Estado | B1000);
        Exit = false; Salir = true;
    }
    break; //

    case 2: Salir = true; break; //Se seleccionó la opción 2 (NO)
    }
}

if(!Salir && millis()%500 == 0) // Muestra en pantalla el MENU
{
    lcd.setCursor(1,0);
    lcd.print(F("Agregar Pasajero:"));

```

```

        lcd.setCursor(1,1);
        lcd.print(F("SI"));
        lcd.setCursor(1,2);
        lcd.print(F("NO"));

        for( int i=0 ; i<rowsLCD ; i++ )
        {
            lcd.setCursor(0,i);
            lcd.print(" ");
        }
        lcd.setCursor(0,menuPosition);
    }

    if(millis() - tiempo > 5000) //
    {
        Salir = true;
    }
}

printer.wake();    // MUST wake() before printing again, even if reset
printer.setDefault(); // Restaurar la impresora.
BNP = HIGH;
Salir = true;
lcd.clear();
}

```

/* Función que Muestra el Menú para terminar carreras en el LCD.

* Parámetro Estado: Indica qué pasajeros están abordo, 0: Indica Ausencia de pasajero; 1: Indica Pasajero A Bordo;

* Parámetro d1 : Distancia TOTAL recorrida por el Pasajero 1.

* Parámetro d2 : Distancia TOTAL recorrida por el Pasajero 2.

* Parámetro d3 : Distancia TOTAL recorrida por el Pasajero 3.

```
* Parámetro d4 : Distancia TOTAL recorrida por el Pasajero 4.  
*/
```

```
void Terminar_Carrera()  
{  
    lcd.clear();  
    menuPosition = 1;  
    tiempo = millis();  
  
    while(!Salir)  
    {  
        readButtons();  
        if( btnUp == 0 && menuPosition-2 >= 0 ) //  
        {  
            tiempo = millis();  
            menuPosition--;  
        }  
        else if( btnDown == 0 && menuPosition+1 < menuOptions )//  
        {  
            tiempo = millis();  
            menuPosition++;  
        }  
        else if (btnCenter == 0) //  
        {  
            switch(menuPosition) //  
            {  
                case 1: if((Estado & B0001) == B0001){ Imprimir_Recibo(1, d1,  
distancia1); d1 = 0; distancia1 = 0; Estado = (Estado & B1110); } Salir = true;  
break;
```

```
        case 2: if((Estado & B0010) == B0010){ Imprimir_Recibo(2, d2,
distancia2); d2 = 0; distancia2 = 0; Estado = (Estado & B1101); } Salir = true;
break;
```

```
        case 3: if((Estado & B0100) == B0100){ Imprimir_Recibo(3, d3,
distancia3); d3 = 0; distancia3 = 0; Estado = (Estado & B1011); } Salir = true;
break;
```

```
        case 4: if((Estado & B1000) == B1000){ Imprimir_Recibo(4, d4,
distancia4); d4 = 0; distancia4 = 0; Estado = (Estado & B0111); } Salir = true;
break;
```

```
        case 5:                                     Salir = true;
break;
    }
}
```

```
if( millis()%500 == 0 ) //
{
    if( menuPosition <= 3 ) //
    {
        for( int i=0; i< 4 ; i++ )
        {
            lcd.setCursor(1, i % rowsLCD);
            lcd.print( i<menuOptions ? nMENU[i] : "          " );
        }
    }
}
```

```
if( menuPosition % rowsLCD == 0 ) // Segunda parte del Menú
{
    for( int i=menuPosition ; i<(menuPosition+rowsLCD) ; i++ )
    {
        lcd.setCursor(1, i % rowsLCD);
        lcd.print( i<menuOptions ? nMENU[i] : "          " );
    }
}
```

```

        }
    }

    for( int i=0 ; i<rowsLCD ; i++ )
    {
        lcd.setCursor(0,i);
        lcd.print(" ");
    }
    lcd.setCursor(0, menuPosition % rowsLCD );
}

if(millis() - tiempo > 5000)
{
    Salir = true;
}
}

printer.setDefault(); // Restaurar impresora.
Salir = true;
BSTOP = HIGH;
lcd.clear();
}

/*****
/***** FUNCION QUE PERMITE IMPRIMIR EL TICKET *****/
*****/

void Imprimir_Ticket(uint8_t Pasajero)
{
    menuPosition = 1;
    lcd.clear();

```

```
tiempo = millis();
```

```
while(!Salir)
```

```
{
```

```
    readButtons();
```

```
    if (!btnDown && menuPosition == 1)
```

```
    {
```

```
        tiempo = millis();
```

```
        menuPosition=2;
```

```
    }
```

```
    else if (!btnUp && menuPosition == 2)
```

```
    {
```

```
        tiempo = millis();
```

```
        menuPosition=1;
```

```
    }
```

```
    else if(!btnCenter )
```

```
    {
```

```
        switch( menuPosition )
```

```
        {
```

```
            case 1:
```

```
                lcd.clear();
```

```
                PORTC = (1 << Led_Verde) | (1 << Led_Rojo) | (0 << Led_Azul); //
```

```
                lcd.setCursor(0, 0);
```

```
                lcd.print(F("PASAJERO"));
```

```
                lcd.setCursor(0,1);
```

```
                lcd.print(F("AGREGADO:"));
```

```
                lcd.setCursor(0, 3);
```

```
                lcd.print(F("IMPRIMIENDO..."));
```

```
                printer.inverseOn();
```

```
                printer.doubleHeightOn();
```

```
                printer.justify('C');
```

```
printer.setSize('L');
printer.println(F("UNI"));
printer.inverseOff();
printer.justify('C');
printer.setSize('L');
printer.print(F("PASAJERO: "));
printer.println(Pasajero);
printer.println();
printer.println();
printer.println();
Salir = true;
break;
```

case 2:

```
lcd.clear();
tiempo = millis();
while (millis() - tiempo < 3500)
{
    lcd.setCursor(0,0);
    lcd.print(F("PASAJERO"));
    lcd.setCursor(0,1);
    lcd.print(F("AGREGADO:"));
    lcd.setCursor(0,3);
    lcd.print(F("ROJO"));
}
Salir = true;
break; //
}
```

```
if(!Salir && millis()%500 == 0)
```



```

    {
        lcd.setCursor(1,0); lcd.print(F("Imprimir Ticket:"));
        lcd.setCursor(1,1); lcd.print(F("SI"));
        lcd.setCursor(1,2); lcd.print(F("NO"));
        for( int i=0 ; i<rowsLCD ; i++ )
        {
            lcd.setCursor(0,i);
            lcd.print(" ");
        }
        lcd.setCursor(0,menuPosition);
    }

    if(millis() - tiempo > 5000)
    {
        lcd.setCursor(0,0);
        lcd.print(F("PASAJERO"));
        lcd.setCursor(0,1);
        lcd.print(F("AGREGADO:"));
        lcd.setCursor(0,3);
        lcd.print(F("ROJO"));
        Salir = true;
    }
}

PORTC = (0 << Led_Verde) | (1 << Led_Rojo) | (1 << Led_Azul); //
}

/*****
/***** FUNCION QUE PERMITE IMPRIMIR EL RECIBO *****/
/*****/

void Imprimir_Recibo(uint8_t Pasajero, float d, float distancia)
{

```

```

menuPosition = 1;
lcd.clear();
tiempo = millis();
BSTOP = HIGH;
int Precio;
//PORTC = 0x0E; // Leds y buzzer apagados
while(!Salir)
{
    Precio = Calcular_Precio(distancia);
    readButtons();
    if (!btnDown && menuPosition == 1)
    {
        tiempo = millis();
        menuPosition=2;
    }
    else if (!btnUp && menuPosition == 2)
    {
        tiempo = millis();
        menuPosition=1;
    }
    else if(!btnCenter )
    {
        switch( menuPosition )
        {
            case 1:
                lcd.clear();
                PORTC = (1 << Led_Verde) | (1 << Led_Rojo) | (0 << Led_Azul);
//
                lcd.setCursor(0,3); lcd.print(F("IMPRIMIENDO"));
                lcd.setCursor(0, 0);
                lcd.print(F("A PAGAR"));

```

```
lcd.setCursor(18, 2);
lcd.print(F("C$"));
// Test inverse on & off
printer.inverseOn();
printer.doubleHeightOn();
printer.justify('C');
printer.setSize('L');
printer.println(F("UNI"));
printer.inverseOff();
printer.doubleHeightOff();
printer.println();
printer.setSize('M');
printer.print(F("Pasajero "));
printer.println(Pasajero);
printer.println();
printer.justify('L');
printer.setSize('S');
printer.print(F("Fecha Emision    : "));
if (day() < 10)
{
printer.print("0");
printer.print(day());
}
else
{
printer.print(day());
}
printer.print("/");
if (month() < 10)
{
printer.print("0");
```

```
printer.print(month());
}
else
{
printer.print(month());
}
printer.print("/");
printer.println(year());
printer.print(F("Hora Emision    : "));
if (hour() < 10)
{
printer.print("0");
printer.print(hour());
}
else
{
printer.print(hour());
}
printer.print(":");
if (minute() < 10)
{
printer.print("0");
printer.print(minute());
}
else
{
printer.print(minute());
}
printer.print(":");
if (second() < 10)
{
```

```

        printer.print("0");
        printer.println(second());
    }
    else
    {
        printer.println(second());
    }
    printer.print(F("Tarifa Diu/Noc[1/2]: "));
    printer.println(Tarifa);
    printer.print(F("Dist. Recorrida[Km]: "));
    printer.println(d,1);
    printer.println();
    printer.setSize('S');
    printer.println(F("....."));
    printer.setSize('M');
    printer.print(F(" Valor a Pagar [C$]: "));
    printer.println(Precio);
    printer.setSize('S');
    printer.println(F("....."));
    printer.println();
    printer.println();
    printer.println();
    Salir = true;
break;

```

case 2:

```

    lcd.clear();
    tiempo = millis();
    while (BSTOP)
    {
        lcd.setCursor(0, 0);

```

```

    lcd.print(F("A PAGAR"));
    lcd.setCursor(0, 2);
    lcd.print(F("DIST:"));
    lcd.setCursor(1, 3);
    lcd.print(d, 1);
    lcd.setCursor(4, 3);
    lcd.print(F("Km"));
    lcd.setCursor(18, 2);
    lcd.print(F("C$"));
    if(millis()- tiempo > 30000)
    {
        PORTD = (1 << led_lcd) | (1 << buzzer); // buzzer encendido
        _delay_ms(50);
        PORTD = (1 << led_lcd) | (0 << buzzer); // buzzer apagados
        _delay_ms(500);
    }
    Salir = true;
    break; //
}
}

```

```

if(!Salir && millis()%500 == 0)
{
    lcd.setCursor(1,0); lcd.print(F("Imprimir Recibo:"));
    lcd.setCursor(1,1); lcd.print(F("SI"));
    lcd.setCursor(1,2); lcd.print(F("NO"));
    for( int i=0 ; i<rowsLCD ; i++ )
    {
        lcd.setCursor(0,i);
        lcd.print(" ");
    }
}

```

```

        lcd.setCursor(0,menuPosition);
    }
    if(!Salir && millis() - tiempo > 5000)
    {
        lcd.clear();
        tiempo = millis();
        while (BSTOP)
        {
            lcd.setCursor(0, 0);
            lcd.print(F("A PAGAR"));
            lcd.setCursor(0, 2);
            lcd.print(F("DIST:"));
            lcd.setCursor(1, 3);
            lcd.print(d,1);
            lcd.setCursor(4, 3);
            lcd.print(F("Km"));
            lcd.setCursor(18, 2);
            lcd.print(F("C$"));
            if(millis()- tiempo > 30000)
            {
                PORTD = (1 << led_lcd) | (1 << buzzer); // buzzer encendido
                _delay_ms(50);
                PORTD = (1 << led_lcd) | (0 << buzzer); // buzzer apagados
                _delay_ms(500);
            }
            Salir = true;
        }
    }
    PORTC = (0 << Led_Verde) | (1 << Led_Rojo) | (1 << Led_Azul); //
    BSTOP = HIGH;
}

```

```

    }
}
/*
 * Función que mide la distancia recorrida, tomada del GPS.
 */
float Sensor_Odometro()
{
    if (gps.location.isValid())
    {
        lat1 = (gps.location.lat());
        lon1 = (gps.location.lng());
    }
    smartDelay(850);
    if (gps.location.isValid())
    {
        lat2 = (gps.location.lat());
        lon2 = (gps.location.lng());
    }
    distance = (dist(lat1, lon1, lat2, lon2)/1000);
    if(distance > 0.05)
    {
        distance = 0;
    }
}

int Calcular_Precio(float d_Recorrida)
{
    uint8_t TI1 = 19;
    uint8_t TI2 = 19;
    float Total_Km;
    float PxKm_1 = 6;

```



```
float PxKm_2 = 6;
```

```
int Precio;
```

```
Total_Km = (d_Recorrida * PxKm_1) + TI1; // P_Km precio totl x istnci recorri
```

```
if(Total_Km > 25)
```

```
    Precio = round (Total_Km); // TI1 Trif inicil 1
```

```
else
```

```
    Precio = 25;
```

```
if(Precio < 100)
```

```
{
```

```
    if((Precio%5)>=2.5)
```

```
    {
```

```
        Precio = Precio + 5 - (Precio%5);
```

```
    }
```

```
    else
```

```
        Precio = Precio - (Precio%5);
```

```
}
```

```
else if(Precio >= 100)
```

```
{
```

```
    if(((Precio%100)%5)>=2.5)
```

```
    {
```

```
        Precio = Precio + 5 - ((Precio%100)%5);
```

```
    }
```

```
    else
```

```
        Precio = Precio - ((Precio%100)%5);
```

```
}
```

```
return(Precio);
```

```
}
```

```
// *****  
  
//                               SUBROUTINAS  
  
// *****
```

```
static void smartDelay(unsigned long ms)
```

```
{  
    tiempo = millis();  
    do  
    {  
        while (Gps.available())  
            gps.encode(Gps.read());  
    } while (millis() - tiempo < ms);  
}
```

```
// Retorna la distancia en metros entre 2 posiciones
```

```
float dist(float lat1, float long1, float lat2, float long2)
```

```
{  
    float delta = radians(long1-long2);  
    float sdlong = sin(delta);  
    float cdlong = cos(delta);  
    lat1 = radians(lat1);  
    lat2 = radians(lat2);  
    float slat1 = sin(lat1);  
    float clat1 = cos(lat1);  
    float slat2 = sin(lat2);  
    float clat2 = cos(lat2);  
    delta = (clat1 * slat2) - (slat1 * clat2 * cdlong);  
    delta = sq(delta);  
    delta += sq(clat2 * sdlong);  
    delta = sqrt(delta);  
}
```

```

float denom = (slat1 * slat2) + (clat1 * clat2 * cdlong);
delta = atan2(delta, denom);
return delta * 6372795;
}

```

```

void Nuevo_Pasajero()
{
    BNP = LOW;
    sleep_disable();//desactivamos el modo bajo consumo
}

```

```

void Detener_Carrera()
{
    BSTOP = LOW;
    sleep_disable();//desactivamos el modo bajo consumo
}

```

```

void readButtons()
{
    btnDown  = PINC & (1<<PC0); // Boton que mueve el cursor hacia abajo
    btnUp    = PINC & (1<<PC1); // Boton que mueve el cursor hacia arriba
    btnCenter = PINC & (1<<PC2); // Boton que ingresa en la opción

    if(!btnDown || !btnUp || !btnCenter )
    {
        while(!(PINC & (1<<PC0)) || !(PINC & (1<<PC1)) || !(PINC & (1<<PC2)))
        {
            digitalWrite(buzzer,HIGH);
            _delay_ms(30);
            digitalWrite(buzzer,LOW); // buzzer apagados
        }
    }
}

```

}

ANEXO 2 – LEY GENERAL DE TRANSPORTE TERRESTRE

publicado en la Gaceta No. 72 del 14 de abril del 2005

EL PRESIDENTE DE LA REPÚBLICA DE NICARAGUA

En uso de las facultades que le confiere la Constitución Política,

HA DICTADO

El siguiente

DECRETO

REGLAMENTO DE LEY GENERAL DE TRANSPORTE TERRESTRE

CAPÍTULO I

DISPOSICIONES GENERALES

Objetivo

Artículo 1.- El presente Reglamento tiene por objeto, establecer las disposiciones administrativas y técnicas para una mejor comprensión y aplicación de la Ley General de Transporte Terrestre (LGTT), publicada en La Gaceta # 72 del jueves 14 de abril del año 2005.

Aplicación

Artículo 2.- La vigilancia y aplicación de la Ley General de Transporte Terrestre (LGTT) y de este Reglamento, corresponde al MTI, a través de la Dirección General de Transporte Terrestre (DGTT), o al organismo administrativo que las municipalidades constituyan para tal efecto.

TÍTULO I

CAPÍTULO II

ASPECTOS COMUNES A LA PRESTACIÓN DEL SERVICIO

Transporte Interurbano

Artículo 93.- Es el traslado de pasajeros bajo la clasificación de transporte escolar o especial, que se realiza entre dos ciudades o poblados de un mismo municipio.

Normas Básicas

Artículo 94.- El servicio de transporte especial debe operar bajo las siguientes normas básicas:

1. Estos servicios están destinados a resolver única y exclusivamente problemas específicos de transporte de personal de empresas e instituciones, así como el transporte de estudiantes, para lo cual deberán estar debidamente autorizados por las municipalidades.
2. Debe establecerse recorridos fijos y lugares de paradas diferentes a las usadas por el Transporte Público.
3. La programación de viajes estará en función de las demandas específicas transporte para las que se autoriza.
4. Las unidades destinadas para este tipo de servicio deben reunir, al menos, los requisitos de calidad exigidos para el transporte público ordinario.

CAPÍTULO XVI TAXIS LOCALES

Servicio

Artículo 95.- Es un servicio selectivo que utiliza un determinado numero de personas sin rutas previamente establecidas, se prestará mediante operación continua, en tiempos u horarios determinados mediante turnos, o cualquier otra

3. Moto Taxi

Vehículos

Artículo 96.- El servicio se prestará con automóviles, del tipo sedán o station wagon, de cuatro puertas, con capacidad de cuatro ó cinco personas, incluido conductor, con una cilindrada mínima de 1500 cc en caso de vehículos de gasolina y 1800 cc en caso de los vehículos de diesel.

En algunas municipalidades podrá autorizarse el servicio de moto taxis. Los vehículos deberán portar en la parte superior la identificación del tipo de servicio que presta, con iluminación para una mejor identificación

Rotulación

Artículo 97.- La rotulación en las puertas delanteras, deberá incluir como mínimo; el tipo de servicio, municipio donde opera, número de placas y organización a la que pertenece, y en la parte posterior deberá reflejarse únicamente el número de placas. En cada municipio se implementará el diseño de una raya de colores a lo largo de las partes medias laterales de los vehículos, y en el caso que se opere por turnos, se establecerá por combinaciones diferentes para cada turno.

Viaje

Artículo 98.- En el trayecto del viaje o “carrera” se inicia en el momento del abordaje y termina en el lugar que solicite el usuario, siempre que dicho lugar sea accesible.

El conductor priorizará el destino de los primeros pasajeros que abordaron el taxi, aunque en el trayecto podrá subir otros pasajeros, siempre y cuando vayan en la misma ruta de los primeros usuarios.

autoridades municipales.

Paradas

Artículo 101.- Los taxis no podrán subir o bajar pasajeros en las paradas o bahías de los autobuses del transporte intramunicipal e intermunicipal, sino a doscientos metros de distancia y cumpliendo con el arto 136 de la ley # 43.

Permiso Especial

Artículo 102.- En aquellos municipios donde no exista el servicio de taxis interlocales, los taxis locales o ruleteros deberán obtener un permiso especial del MTI para poder realizar viajes expresos de particulares de un municipio a otro, salvo el lapso de tiempo entre las 7 pm y 5 am del día siguiente.

Moto Taxi

Artículo 103.- Vehículo automotor tipo motoneta, con capacidad máxima por dos asientos además del conductor, cuando disponga de asiento lateral.

Taxis de Parada

Artículo 104.- Los taxis de parada operarán desde aquellas bahías o paradas autorizadas por las municipalidades en lugares específicos como hoteles, universidades y centros de estudio, mercados, terminales de transporte intermunicipal y centros comerciales. No podrán trasladar pasajeros fuera del casco urbano del municipio que los autorizó.

TÍTULO V

TRANSPORTE DE CARGA

CAPÍTULO XVII

DEFINICIONES GENERALES

Disposiciones Técnicas

Artículo 105.- Los vehículos automotores de transporte de carga, como

ANEXO 3 – DATASHEET ATMEGA 328P

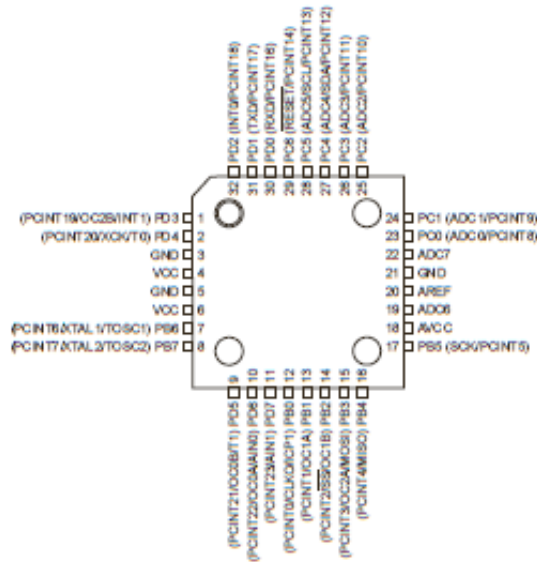
- 131 Powerful Instructions – Most Single Clock Cycle Execution
- 32 x 8 General Purpose Working Registers
- Fully Static Operation
- Up to 20 MIPS Throughput at 20 MHz
- On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
 - 4K/16/32K Bytes of In-System Self-Programmable Flash program memory (ATmega48PA/88PA/168PA/328P)
 - 256/512/512/1K Bytes EEPROM (ATmega48PA/88PA/168PA/328P)
 - 512/1K/1K/2K Bytes Internal SRAM (ATmega48PA/88PA/168PA/328P)
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - 8-channel 10-bit ADC in TQFP and QFN/MLF package
 - Temperature Measurement
 - 6-channel 10-bit ADC in PDIP Package
 - Temperature Measurement
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Byte-oriented 2-wire Serial Interface (Philips I²C compatible)
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
 - 23 Programmable I/O Lines
 - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
 - 1.8 - 5.5V for ATmega48PA/88PA/168PA/328P
- Temperature Range:
 - -40°C to 85°C
- Speed Grade:
 - 0 - 20 MHz @ 1.8 - 5.5V
- Low Power Consumption at 1 MHz, 1.8V, 25°C for ATmega48PA/88PA/168PA/328P:
 - Active Mode: 0.2 mA
 - Power-down Mode: 0.1 µA
 - Power-save Mode: 0.75 µA (Including 32 kHz RTC)



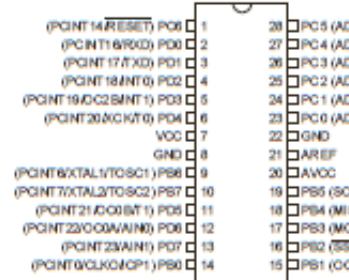
8-bit AVR[®]
Microcontroller
 with 4K/16/32K
 Bytes In-System
 Programmable
 Flash

ATmega48PA
 ATmega88PA
 ATmega168PA
 ATmega328P

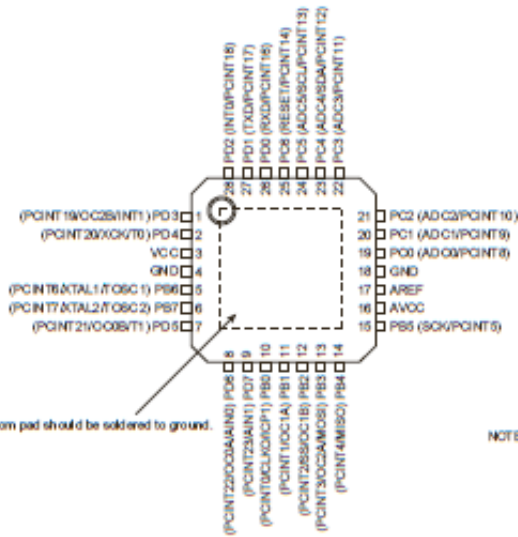
TQFP Top View



PDIP

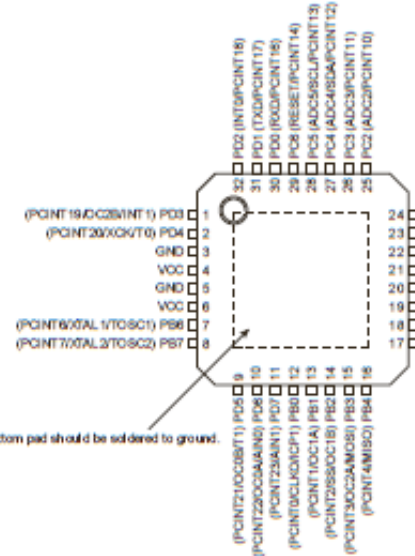


28 MLF Top View



NOTE: Bottom pad should be soldered to ground.

32 MLF Top View



NOTE: Bottom pad should be soldered to ground.

ANEXO 4 – DATASHEET ATMEGA 8U2

- High Performance, Low Power AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
 - 125 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16 MHz
- Non-volatile Program and Data Memories
 - 8K/16K/32K Bytes of In-System Self-Programmable Flash
 - 512/512/1024 EEPROM
 - 512/512/1024 Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/ 100,000 EEPROM
 - Data retention: 20 years at 85°C/ 100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by on-chip Boot Program hardware-activated after reset
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- USB 2.0 Full-speed Device Module with Interrupt on Transfer Completion
 - Complies fully with Universal Serial Bus Specification REV 2.0
 - 48 MHz PLL for Full-speed Bus Operation : data transfer rates at 12 Mbit/s
 - Fully independent 176 bytes USB DPRAM for endpoint memory allocation
 - Endpoint 0 for Control Transfers: from 8 up to 64-bytes
 - 4 Programmable Endpoints:
 - IN or Out Directions
 - Bulk, Interrupt and Isochronous Transfers
 - Programmable maximum packet size from 8 to 64 bytes
 - Programmable single or double buffer
 - Suspend/Resume Interrupts
 - Microcontroller reset on USB Bus Reset without detach
 - USB Bus Disconnection on Microcontroller Request
- Peripheral Features
 - One 8-bit Timer/Counters with Separate Prescaler and Compare Mode (two 8-bit PWM channels)
 - One 16-bit Timer/Counter with Separate Prescaler, Compare and Capture Mode (three 8-bit PWM channels)
 - USART with SPI master only mode and hardware flow control (RTS/CTS)
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- On Chip Debug Interface (debugWIRE)
- Special Microcontroller Features
 - Power-On Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Five Sleep Modes: Idle, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
 - 22 Programmable I/O Lines
 - QFN32 (5x5mm) / TQFP32 packages
- Operating Voltages
 - 2.7 - 5.5V
- Operating temperature
 - Industrial (-40°C to +85°C)
- Maximum Frequency
 - 8 MHz at 2.7V - Industrial range
 - 16 MHz at 4.5V - Industrial range

Note: 1. See "Data Retention" on page 6 for details.

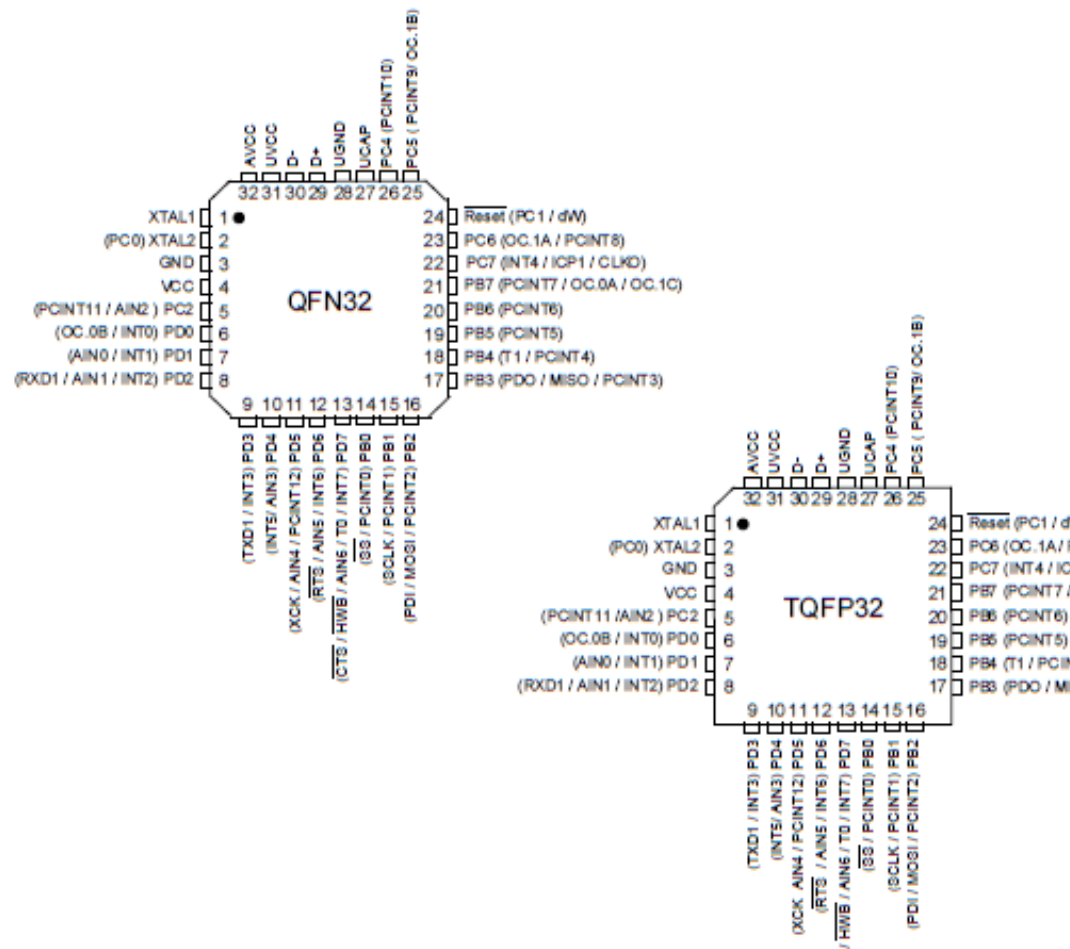


8-bit **AVR®**
Microcontro
with
8/16/32K By
of ISP Flash
and USB
Controller

ATmega8U

ATmega16

ATmega32



Note: The large center pad underneath the QFN package should be soldered to ground to ensure good mechanical stability.

1.1 Disclaimer

Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Minimum values will be available after the device is characterized.

ANEXO 5 – DATASHEET LCD SYSTRONIX

20X4

Here is brief data for the Systronix 20x4 character LCD. It is a Data Vision part and uses the Samsung KS0066 LCD controller. It's a clone of the Hitachi HD44780. We're not aware of any incompatibilities between the two - at least we have never seen any in all the code and custom applications we have done.

This 20x4 LCD is electrically and mechanically interchangeable with 20x4 LCDs from several other vendors. The only differences we've seen among different 20x4 LCDs are:

- 1) LED backlight brightness, voltage and current vary widely, as does the quality of the display
- 2) There is a resistor "Rf" which sets the speed of the LCD interface by controlling the internal oscillator frequency. Several displays we have evaluated have a low resistor value. This makes the display too slow. Looking at the Hitachi data sheet page 56, it appears that perhaps the "incorrect" resistor is really intended for 3V use of the displays.

At 5V the resistor Rf should be 91 Kohms. At 3V it should be 75 Kohms. Using a 3V display at 5V is acceptable from a voltage standpoint (the display can operate on 3-5V) but the oscillator will then be running too slowly. One fix is to always check the busy flag and not use a fixed time delay in your code, then it will work regardless of the LCD speed. The other option is to always allow enough delay for the slower display.

All Systronix 20x4 LCDs have the 91 Kohm resistor and are intended for 5V operation.

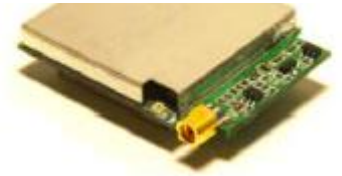
Thank you for purchasing Systronix embedded control products and accessories. If you have any other questions please email to support@systronix.com or phone +1-801-534-1017, fax +1-801-534-1019.

user RAM	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
xxxx0000	CO RAM (1)			00P`P								-	タ	三	α	p	
xxxx0001	(2)			!1AQa4								。	ア	チ	△	ā	q
xxxx0010	(3)			"2BRbr								「	イ	ツ	×	p	θ
xxxx0011	(4)			#3CScs								」	ウ	テ	モ	ε	∞
xxxx0100	(5)			\$4DTdt								、	エ	ト	ト	μ	Ω
xxxx0101	(6)			%5EUeu								・	オ	ナ	1	ε	U
xxxx0110	(7)			&6FUfu								ヲ	カ	ニ	ヨ	p	Σ
xxxx0111	(8)			'7GWgw								フ	キ	ズ	ラ	q	π
xxxx1000	(1)			(8HXhx								ィ	ク	ネ	リ	J	×
xxxx1001	(2)			>9IYiy								お	ケ	ノ	ル	ノ	u
xxxx1010	(3)			*:JZjz								エ	コ	ハ	レ	j	≠
xxxx1011	(4)			+;KLk{								オ	サ	ヒ	ロ	*	斤
xxxx1100	(5)			,<L¥I!								カ	シ	フ	ワ	+	冢
xxxx1101	(6)			-=MIm}								ユ	ズ	ハ	ン	も	÷
xxxx1110	(7)			。>N^n→								ヨ	セ	ホ	°	斤	
xxxx1111	(8)			/?O_oe								ッ	ツ	マ	°	○	■

Note: The user can specify any pattern for character-generator RAM.

ANEXO 6 – DATASHEET GPS PMB-688

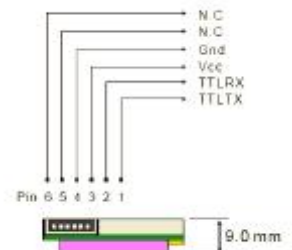
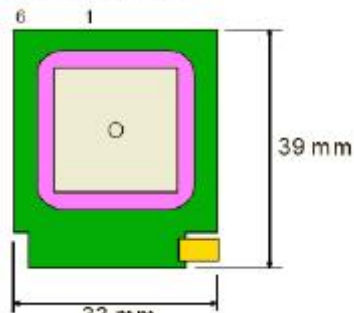
PMB-688 GPS module



PMB-688 FEATURES

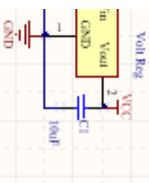
- Built-in SiRFstarIII chipsets receivers give unparalleled GPS performance and precision. 20 parallel satellite-tracking channels for fast acquisition and reacquisition.
- Built-in WAAS/EGNOS Demodulator.
- Low power consumption and ultra mini size only 33x39mm.
- Built-in rechargeable battery for backup memory and RTC backup.
- Support NMEA0183 v2.2 data protocol.
- Enhanced algorithms providing superior navigation performance in urban, canyon and foliage environments.
- For Car Navigation, Marine Navigation, Fleet Management, AVL and Location-Based Services, Auto Pilot, Personal Navigation or touring devices, Tracking devices/systems and Mapping device application.
- Include RF MMCX connector (Optional: Active Antenna)

Connector Pitch 1.25 mm

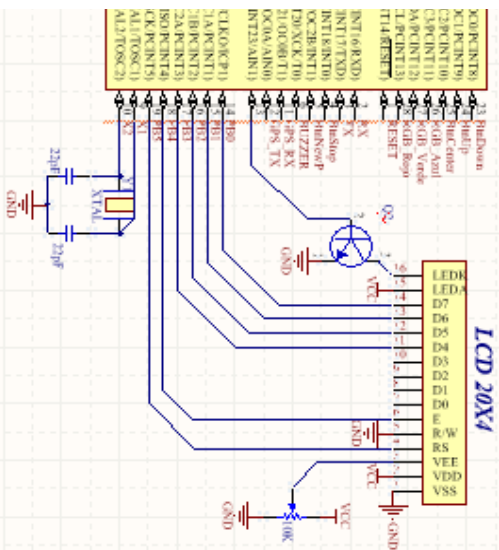


GPS IC	SiRFstar III
Receiver:	Tracking up to 20 satellites L1, 1575.42 MHz, C/A code
Accuracy:	Position: 2DRMS approximately 5m, WAAS support Velocity: 0.1 m/s without SA imposed. Time: $\pm 1\mu\text{sec}$
Acquisition Time:	Cold Start: 42sec (Average) Warm Start: 38sec (Average) Hot Start: 1sec (Min.)
Sensitivity:	Acquisition: -148dBm Tracking: -159 dBm
Dynamics:	Altitude: 18000m (Max.) Velocity: 515m/s (Max.) Acceleration: $\pm 4g$ (Max.)
Navigation update rate :	Once per second
Serial Port :	TTL
Baud Rate :	4800 bps (Optional 9600,19300,38400 bps)
Output Message:	NMEA0183 V2.2 GGA, GSV, GSA, RMC (optional VTG, GLL)
Datum:	WGS 84
Power supply :	DC 3.3V ~ 5V
Power Consumption :	Typical 65mA @12V
LED Function	Power on/off and Navigation
Operating Temp.:	-20°C~+70°C
Storage Temp.:	-30°C~+85°C
Humidity:	5%~95%

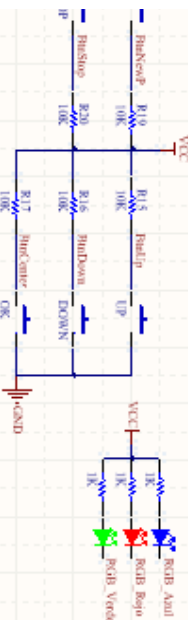
ANEXO 7 –DIAGRAMA ESQUEMATICO FINAL



CONTROLADO



BOTONES DE CONTROL



LED RGB



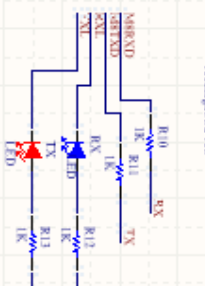
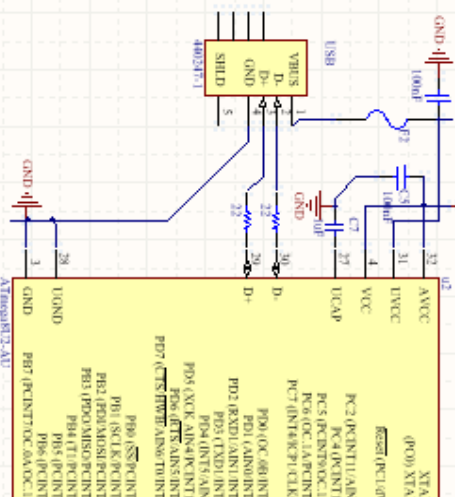
BUZZER



GPS

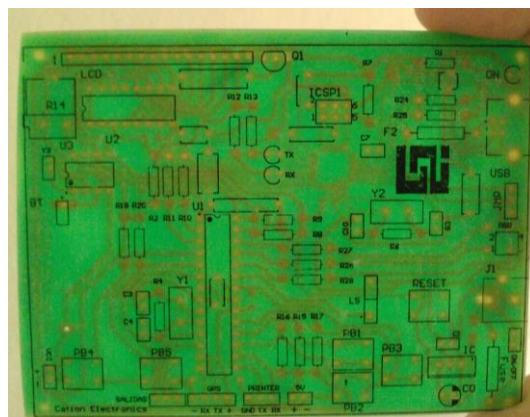


IMPRESOR

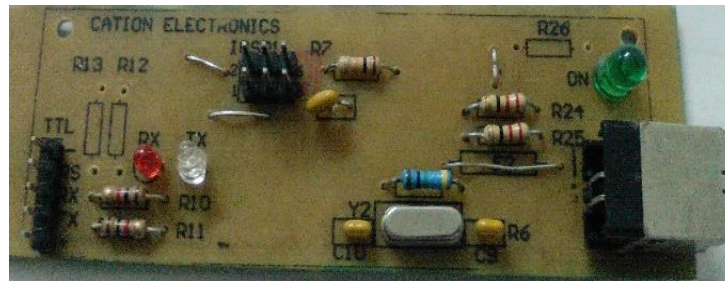


ANEXO 8 –PLACAS DEL CIRCUITO IMPRESO

ANEXO 8.1- PLACA PRINCIPAL



ANEXO 8.2- PLACA USB, VISTA FRONTAL



ANEXO 8.3- PLACA USB, VISTA TRASERA

